

34615.

KARIŞIK MODELLİ DETERMİNİSTİK MONTAJ HATTI Dengeleme
PROBLEMLERİ İÇİN YENİ MODELLER

34615

DOKTORA TEZİ
(ENDÜSTRİ MÜHENDİSLİĞİ)

Hadi GÖKÇEN

Eylül - 1994

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bu tezin DOKTORA tezi olarak uygun olduğunu onaylarım.

Erdal Erel

Danışman

Doç. Dr. Erdal EREL

Sınav Jürisi

Başkan : Prof. Dr. Fevzi KUTAY

Üye : Doç. Dr. Erdal EREL

Üye : Doç. Dr. Serpil EROL

Stokam
Erdal Erel
Serpil Erol

Bu Tez Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Esaslarına Uygundur.

S. Çamur

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	ii
TEŞEKKÜR	iii
KULLANILAN NOTASYON VE PARAMETRELERİN TANIMI	iv
TABLOLARIN LİSTESİ	vi
ŞEKİLLERİN LİSTESİ	viii
EKLERİN LİSTESİ	ix
 GİRİŞ	
 BÖLÜM 1	
MONTAJ HATTI DENGEMELEME PROBLEMİ	5
1.1. Üretim Sistemleri	5
1.2. Seri (Akış Tipi) Üretimin Karakteristikleri	7
1.3. Seri (Akış Tipi) ve Akış Hattı Üretim Sistemleri için Ön Şartlar	9
1.4. Üretim Akış Hatları	10
1.5. Tanımlar ve Terminoloji	13
1.6. Montaj Hattı Dengeleme Problemi ve Sınıflandırılması	18
1.6.1. Tek modellenli deterministik (TMD) hat dengeleme problemi	22
1.6.2. Tek modellenli stokastik (TMS) hat dengeleme problemi	24
1.6.3. Çok/Karışık modellenli deterministik (ÇMD) hat dengeleme problemi	26

1.6.4. Çok/Karışık modelli stokastik (ÇMS) hat dengeleme problemi	30
1.7. Çalışmanın Amacı	31
BÖLÜM 2	
KAYNAK ARAŞTIRMASI	33
2.1. Tek Modelli Deterministik Hat Dengeleme Problemi (TMD)	33
2.2. Tek Modelli Stokastik Hat Dengeleme Problemi (TMS)	38
2.3. Çok/Karışık Modelli Deterministik Hat Dengeleme Problemi (ÇMD)	42
2.4. Çok/Karışık Modelli Stokastik Hat Dengeleme Problemi (ÇMS)	45
BÖLÜM 3	
DETERMINİSTİK KARIŞIK MODELLİ MHD PROBLEMİ İÇİN OPTİMAL ALGORİTMALAR	46
3.1. Roberts ve Villa 'nın (1970) Tamsayılı Programlama Modeli	47
3.2. Geliştirilen Yeni 0-1 Tamsayılı Programlama Modeli	51
3.2.1. Örnek problem	57
3.3. Modellerin Karşılaştırılması	61
3.4. En Kısa Yol Algoritması	62
3.4.1. Durumlar ya da düğümlerin üretilmesi	63
3.4.2. Yönlü okların yapılanması	64
3.4.3. Örnek problem	64
3.5. Geliştirilen Serim Modeli	68

3.5.1. Durumların Üretilmesi	71
3.5.2. Örnek problem	73
3.5.3. Modelin davranışları	76
BÖLÜM 4	
KARIŞIK MODELLİ MONTAJ HATLARININ DENGELENMESİ İÇİN BİR SEZGİSEL YÖNTEM	80
4.1. Sezgisel Metodun Adımları	84
4.2. Hesaplama Sonuçları	88
4.3. Sonuçların Değerlendirilmesi	95
BÖLÜM 5	
KARIŞIK MODELLİ MHD PROBLEMLERİ İÇİN AMAÇ PROGRAMLAMA YAKLAŞIMI	97
5.1. ÇMD Problemin Amaç Programlama Modeli	99
5.2. Modelin Örnek Problemlerle Denenmesi ve Öncelik Yapılarının Duyarlılığının Araştırılması	109
SONUÇ VE TARTIŞMA	115
KAYNAKLAR	119
EKLER	128
ÖZGEÇMİŞ	187

KARIŞIK MODELLİ DETERMİNİSTİK MONTAJ HATTI DENGEME

PROBLEMLERİ İÇİN YENİ MODELLER

(Doktora Tezi)

Hadi GÖKÇEN

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Eylül 1994

ÖZ

Bu çalışmada, karışık modellenli deterministik montaj hattı dengeleme (MHD) problemleri incelenmiştir. Literatürde bu problemlerle ilgili optimal çözüm teknikleri analiz edilerek alternatif modeller önerilmiştir. Yine aynı tür problemler için bir sezgisel metot geliştirilmiştir. Sezgisel metot, 10, 20, 30, 40, 60 görevli ve farklı esneklik oranlı (E-Oranlı) test problemleri için denenmiş ve metodun davranışları incelenmiştir. Çalışmada son olarak, yöneticiye birden fazla amacı gerçekleştirme imkanı vermesi ve literatürde şu ana kadar üzerinde çalışılmamış olması bakımından, çok amaçlı karar verme tekniklerinden Amaç programlama metodu kullanılmış ve yeni bir 0-1 Amaç programlama modeli geliştirilmiştir. Bu tez çalışmasındaki 0-1 formülasyonlar GAMS paket programı ile modellenerek çözülmüş, sezgisel metodun yazılımının kodlanmasında ise FORTRAN dili kullanılmıştır.

NEW MODELS FOR DETERMINISTIC MIXED-MODEL
ASSEMBLY LINE BALANCING PROBLEMS
(Ph.D. Thesis)

Hadi GÖKÇEN
GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
SEPTEMBER 1994

ABSTRACT

In this study, mixed-model deterministic assembly line balancing problems are investigated. Currently available optimal solution techniques for mixed-model assembly line balancing problems reported in the literature are analyzed and alternative models are suggested to solve the problem. In addition, a heuristic solution procedure is also developed and tested on problems with 10, 20, 30, 40 and 60 tasks and various F-ratios. Finally, a 0-1 Goal programming model of the problem is constructed; the goal programming model provides multiobjectives to the decision maker. In the dissertation, the binary models are constructed and solved with GAMS package program; the heuristic procedure is coded in FORTRAN language.

TEŞEKKÜR

Tez çalışmamın başından sonuna kadar benden hiç bir yardımı esirgemeyen ve değerli görüş ve katkılarıyla beni yönlendiren tez hocam Sayın Doç. Dr. Erdal EREL'e, çalışmama gösterdiği hoşgörü ve anlayışı ile manevi desteğini sürekli hissettiğim Bölüm Başkanımız ve Fen Bilimleri Enstitüsü Müdürü Sayın Prof. Dr. Yalçın EROL 'a ve çalışmam boyunca hiç bir fedakarlığı esirgemeyen sevgili eşim Songül GÖKÇEN'e teşekkürü bir borç bilirim.



KULLANILAN NOTASYON VE PARAMETRELERİN TANIMI

- N : Toplam görev sayısı $i=1, \dots, N$
- K : Toplam iş istasyon sayısı $k=1, \dots, K$
- S_k : k istasyonunun zamanı
- P : Ürün (model) sayısı $m=1, \dots, P$
- C : Çevrim zamanı
- C_m : m ürününün çevrim zamanı
- D : Talep hızı
- t_i : i görevinin yapılması için gerekli zaman
- t_{im} : m ürününün i görevini yapmak için gerekli zaman
- Y : Öncelik matrisindeki sıfırların sayısı
- E_i : i görevinin atanabileceği en erken istasyon sayısı
- L_i : i görevinin atanabileceği en geç istasyon sayısı
- MS_i : i düğümündeki model(ürün)lerin seti (i görevine sahip model(ürün)lerin seti).
- P_i : Birleştirilmiş öncelik diyagramında, i görevinden (düğüm) önce gelen tüm görevlerin seti.
- S_i : Birleştirilmiş öncelik diyagramında, i görevini takip eden tüm görevlerin seti.
- W_{km} : m modelinin k istasyonuna atanabilen tüm görevlerinin seti. (Bu set, EK-5 'deki Program WKM 'den elde edilebilir).
- $|W_{km}|$: W_{km} setindeki görevlerin toplam olarak sayısı
- PA_i : Birleştirilmiş öncelik diyagramındaki i düğümünün pozisyon ağırlığı
- $C_{müs}$: m modelinin (ürününün) çevrim zamanı üst sınırı

C_{mas} : m modelinin çevrim zamanı alt sınırı

US : d_m^+ sapma değişkeninin üst sınır değeri

USN : d_{km}^- sapma değişkeninin üst sınır değeri

S : Arzu edilen iş istasyon sayısı sınır değeri

b_k : k istasyonunun maliyeti

B : İstasyonların kurulması için planlanmış bütçe

P_s : s önceliğine atanan öncelikli ağırlık

V_{ik} : $\begin{cases} 1 & \text{Eğer birleştirilmiş diyagramda i görevi k} \\ & \text{istasyonuna atanmışsa} \\ 0 & \text{Diğer durumlarda.} \end{cases}$

X_{km} : $\begin{cases} 1 & \text{Eğer herhangi bir m modeline ait bir görev k} \\ & \text{istasyonuna atanmışsa,} \\ 0 & \text{Diğer durumlarda.} \end{cases}$

A_k : $\begin{cases} 1 & \text{Eğer k istasyonu bütün ürünler tarafından} \\ & \text{kullanılıyorsa, (ya da tüm } X_{km} \text{ ler k} \\ & \text{istasyonu için 1 e eşitse)} \\ 0 & \text{Diğer durumlarda.} \end{cases}$

TABLOLARIN LİSTESİ

<u>Tablo</u>	<u>Sayfa</u>
Tablo 3.1 Örnek Problem için Optimal İstasyon Atamaları	60
Tablo 3.2 Farklı Büyüklükteki Problemler için iki Modelin Karşılaştırması	61
Tablo 3.3 Durum Üretimi	66
Tablo 3.4 Optimal İstasyon Atamaları	68
Tablo 3.5 Durumların Üretilmesi	74
Tablo 3.6 En Kısa Yol Hesaplamaları	74
Tablo 3.7 Optimal İstasyon Atamaları	75
Tablo 3.8 Modelin Durum Sayıları Bakımından Analizi	78
Tablo 4.1 Öncelik ve Pozisyon Ağırlıkları Tablosu	83
Tablo 4.2 10 Elemanlı Problemler için, iki Metodun İstasyon Sayısı ve Bilgisayar Zamanı Bakımından Karşılaştırılması	90
Tablo 4.3 20 Elemanlı Problemler için, iki Metodun İstasyon Sayısı ve Bilgisayar Zamanı Bakımından Karşılaştırılması	91
Tablo 4.4 30 Elemanlı Problemler için, iki Metodun İstasyon Sayısı ve Bilgisayar Zamanı Bakımından Karşılaştırılması	92
Tablo 4.5 40 Elemanlı Problemler için, iki Metodun İstasyon Sayısı ve Bilgisayar Zamanı Bakımından Karşılaştırılması	93

Tablo 4.6	60 Elemanlı Problemler için, iki Metodun İstasyon Sayısı ve Bilgisayar Zamanı Bakımından Karşılaştırılması	93
Tablo 4.7	10, 20, 30, 40 ve 60 Görevli Problemlerin Çözümü için Gerekli İterasyon sayıları ve Mutlak Tolerans Sınırları	94
Tablo 5.1	Tamsayı ve Amaç Programlama Metotlarının Kullanım Sıklıkları	98
Tablo 5.2	Önceliklerin Duyarlılık Analizi	113
Tablo 5.3	Önceliklerin Duyarlılık Analizi (Detaylı Çevrim Zamanlı)	114

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1 Üretim Sistemleri	6
Şekil 1.2 Çok Modelli Montaj Hattı (4 İstasyonlu)	12
Şekil 1.3 Karışık Modelli Montaj Hattı (4 İstasyonlu)	12
Şekil 1.4 11 Elemanlı bir Öncelik Diyagramı	15
Şekil 1.5 (Şekil 1.4) 'deki Öncelik Diyagramına ait Öncelik Matrisi	16
Şekil 1.6 Montaj Hattı Dengeleme Sistemi	18
Şekil 1.7 Montaj Hattı Dengeleme Problemlerinin Sınıflandırması	21
Şekil 1.8 1. ürün (a) ve 2. ürün (b) 'e ait Öncelik Diyagramları ve Birleştirilmiş Öncelik Diyagramı (c)	29
Şekil 3.1 İki Ürünlü Örnek Problem için Öncelik Diyagramları (a) 1. ürün (b) 2. ürün	58
Şekil 3.2 İki Ürünlü Örnek Problem için Birleştirilmiş Öncelik Diyagramı	59
Şekil 3.3 Öncelik Diyagramları (a) 1.ürün (b) 2.ürün	64
Şekil 3.4 Örnek Problem için Nihai Serim	67
Şekil 3.5 Öncelik Diyagramları (a) 1.ürün (b) 2.ürün (c) Birleştirilmiş	73
Şekil 3.6 Örnek Problem için Nihai Serim	75
Şekil 4.1 İki Ürüne ait Birleştirilmiş Öncelik Diyagramı	82
Şekil 4.2 Geliştirilen Sezgisel Metodun Akış şeması	87

EKLERİN LİSTESİ

- EK-1 Birleştirilmiş Diyagramda, Herbir Görev için E_i ve L_i Sınırlarını Bulan Program EİLi 'nin FORTRAN Diliyle Kodlanmış Yazılımı ve Örnek Bir Problem İçin Çözüm Çıktısı.
- EK-2 Bir Karışık Modelli MHD Probleminin 0-1 Saf (pure) Tamsayılı Programlama Modeli ve Çözüm Çıktısı.
- EK-3 Geliştirilen Sezgisel Yöntemin (Program MixA1b) FORTRAN Diliyle Kodlanmış Yazılımı ve Örnek Bir Problem İçin Çözüm Çıktısı.
- EK-4 İki Ürünlü Karışık MHD Problemi İçin Oluşturulan GAMS Modeli.
- EK-5 Karışık Modelli MHD Problemlerinde, Herbir İstasyona Atanmaya Aday Görevleri Belirleyen Program WKM 'nin FORTRAN Diliyle Kodlanmış Yazılımı ve Örnek Bir Problem İçin Çözüm Çıktısı.
- EK-6 0-1 Amaç Programlama Modeli ((STA) 1. Öncelikli Amaç).
- EK-7 0-1 Amaç Programlama Modeli ((CYC1+CYC2) 2. Öncelikli Amaç).
- EK-8 0-1 Amaç Programlama Modeli ((BOLGE) 3. Öncelikli Amaç).

GİRİŞ

Montaj hatlarında bir ürünün montajı, bir çok parça bileşen ve alt montajın biraraya getirilmesi ve bunların üzerinde birtakım işlemlerin yapılmasıyla gerçekleştirilir. İşlemleri yapacak olanlar, hat boyunca sıralanmış olan işçi grupları ya da diğer bir deyişle iş istasyonlarıdır. Bir montaj hattının temel özelliği, iş parçalarının bir istasyondan diğer bir istasyona hareket etmesidir. Montaj hatları genelde iki farklı tiptedirler; Mekanik olmayan ve hareketli bantlı hatlar. Mekanik olmayan hatlar üzerindeki operatörler mekanik olarak ilerleme etkisinden bağımsızdırlar. Hareketli bantlı hatlar ise, bir konveyör bandıyla karakterize edilmiş ilerleyen hatlardır.

Montaj hatlarının diğer bir sınıflandırması da, üretilen ürünlerin/modellerin sayısıdır. Tek ürünlü hatlarda, tek bir ürünün üretimi söz konusudur. Çok ürünlü hatlarda, iki yada daha fazla benzer tipteki ürünler kümeler halinde ayrı ayrı üretilirken, karışık ürünlü hatlarda, iki ya da daha fazla benzer ürünü aynı anda üretmek mümkündür. Çok ve karışık modelli durumlarda önemli olan, iş yükünün benzer olmasıdır. Bu benzerlik ne kadar fazla olursa, o derece çok modelli veya karışık modelli hat kurulması kolaylaşır.

Mamule göre yerleştirmede, oldukça kısa bir zamanda geniş hacimde üretim yapılmakta ise de, üretim hattı bir

kez kurulduktan sonra, bu tür yerleřtirmede, iřleme gre yerleřtirmede pek o kadar nemli olmayan bir takım problemler ortaya ıkar. Bu olduka karmařık problemlerden birisi de Hat Dengeleme Problemleridir ki, buna iřlemlerin ve istasyonların eřit zamanlar ve istenilen retim oranını karřılamak iin gerekli zamanlar bakımından dengelenmesi problemleri de diyebiliriz.

Montaj hattında meydana gelen bu problemlere czm getirmekteki ama, elemanter iřler arasında bulunan ncelik şartlarını saėlayacak, her istasyondaki toplam iřyk zamanı, verilen evrim zamanından byk olmayacak ve istenilen performans kriteri saėlanacak řekilde, grevlerin iř istasyonlarına atanmasıdır. Elemanların istasyonlara eřit olarak atanması sonucunda boř zamanın (her ne kadar uygulamada tamamen yok edilmesi mmkn olmasa da) minimum seviyeye indirilmesi mmkn olabilecektir. Dolayısıyla boř zamanın getirdiėi maliyet klfetinden kurtulmak da mmkn olacaktır.

Literatrde MHD problemleri iin geliřtirilen birok metot bulunmaktadır. Bu metotların byk bir kısmını Tek modelli deterministik (TMD) hat dengeleme problemleri iin geliřtirilen metotlar oluřturmaktadır. Halbuki birden fazla rnn aynı anda retim hattında retilmesi (Bu tr retime endstride sıka karřılařılmaktadır.) sırasında ortaya ıkan dengeleme problemleri ile ilgili geliřtirilen metotların sayısı yok denecek kadar azdır.

Bu nedenle tez çalışması, Çok modellenli deterministik (ÇMD) MHD problemleriyle ilgili yapılan yeni birtakım araştırmaları kapsamaktadır.

Çalışmanın birinci ve ikinci bölümü sırasıyla, Üretim sistemleri, MHD problemleriyle ilgili genel tanımlamalar ile kaynak araştırmasından oluşmaktadır.

Çalışmanın Üçüncü bölümünde, karışık modellenli MHD problemleri için geliştirilen mevcut optimal algoritmalar açıklanmıştır. Literatürde, ÇMD problemlerle ilgili (ara stokların olmadığı varsayımı altında) optimal metotların denendiği tek çalışma Roberts ve Villa 'ya (1970) aittir. Roberts ve Villa (1970), problemi 0-1 Tamsayılı programlama ve En kısa yol metotlarıyla formüle etmişlerdir. Çalışmanın yine bu bölümünde, birtakım pratik varsayımların ilavesiyle, Roberts ve Villa' nın (1970) optimal metotlarına alternatif iki yeni model sunulmuştur.

Dördüncü bölümde, yine aynı tür problem sınıfı için, Helgeson ve Birnie 'nin (1961) RPW (Ranked Positional Weight) tekniğine dayalı bir sezgisel metot geliştirilmiş ve metodun davranışları incelenmiştir.

Son bölümde ise, ÇMD problemler için, çok amaçlı karar verme tekniklerinden olan Amaç programlama kullanılarak yeni bir 0-1 Amaç programlama modeli geliştirilmiştir.

Ulaşılan sonuçlarla, ileride yapılabilecek çalışmalar için önerilerin bulunduğu sonuç ve tartışma kısmı ile çalışma sona ermektedir.



BÖLÜM 1

MONTAJ HATTI DENGELEME PROBLEMİ

1.1. Üretim Sistemleri

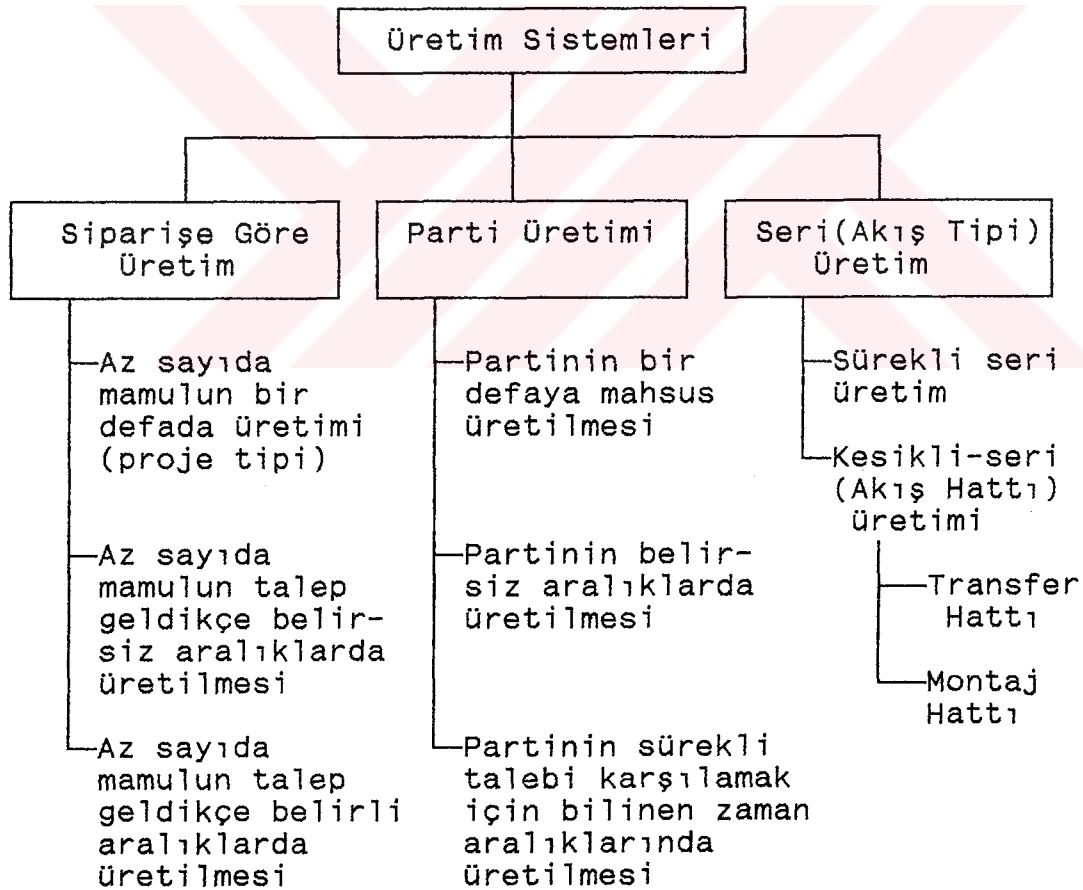
Bir Üretim sistemi, hammadde ve/veya yarımamullerin bir dönüşüm birimiyle ürün haline getirildiği sistem olarak tanımlanabilir. Üretim sistemlerini genel olarak üç başlık altında sınıflandırmak mümkündür (şekil 1.1):

- i) Siparişe göre Üretim
- ii) Parti Üretimi
- iii) Seri (Akış tipi) Üretim.

Siparişe Göre Üretim, üretim miktarı düşük ancak çok fazla çeşitlilik arzeden siparişleri karşılamak üzere yapılan ürettimdir. Doğal olarak, ürün çeşitliliği ve düşük üretim miktarları, işlemlerde tekrarlılığı da en az seviyeye indirmektedir. Bu sistemlerde, bir çok değişik işlemleri yapabilen çok işlemleri tezgahlar kullanılmaktadır.

Parti Üretimi, belirli bir siparişi ya da sürekli talebi karşılamak için benzer veya aynı cinsten ürünlerin partiler halinde üretilmesidir. Bu sistemlerin en büyük özelliği, bir parti bitmeden diğerinin üretimine geçilmemesidir. Ayrıca talep sürekli ve de sipariş tipi üretimde olduğu kadar değişken değildir.

Seri (Akış Tipi) Üretim, talebin büyük olduğu ve ürün çeşitliliğinin az olduğu bir üretim şeklidir. Uzmanlaşmanın gerçekleştirildiği bu üretim sistemlerinin kurulabilmesi için talebin üretim hızından fazla olması şartı vardır. Diğer bir deyişle, üretimin tümü pazar bulabiliyorsa, bu tip bir sistemin kurulması anlam taşır. Aksi halde özel ve pahalı makina ve teçizat gerektiren bu sistemlerde, üretim esnekliği olmadığından, talep düşüşlerinin maliyeti çok yüksek olacaktır.



Şekil 1.1 Üretim Sistemleri

Seri üretimin ana özelliği, ürün akışının olması (ürünün hareket halinde olması) ve tesislerin üretilecek ürüne göre tasarlanmasıdır. Seri (Akış tipi) üretimde, sistemin başlangıç noktasından başlayan hammadde, yarımamul ve parça gibi girdiler, işlem birimlerinden geçer ve son ürün haline dönüşerek sistemden çıkarlar.

1.2. Seri (Akış Tipi) Üretimin Karakteristikleri

Seri üretim, sürekli seri ve kesikli seri (akış hattı) üretim olmak üzere ikiye ayrılır. Sürekli seri üretimde, işlenen hammadde ve ürünler doğal yapıları itibarıyla kendiliğinden akarlar. Çimento, şeker, petrokimya ve gıda maddeleri üretimi bu üretimin en önemli örnekleridir.

Kesikli seri üretimde ürün, tek tek birimler halinde, birbirini takip eden iş istasyonlarındaki gerekli işlemlerin yapılmasıyla oluşur. Motorlu taşıt montaj hatları bu üretim tipinin en güzel örnekleridir. Tesislerin, üretilecek üretimin özelliklerine göre tasarlandığı bu sistemler, teknolojik açıdan en gelişmiş üretim tipini oluştururlar. Bir üretim hattı (Akış hattı), tek bir ürün üretmek ya da aynı ürünün benzer modellerini (farklı işlem sıralı) üretmek için tasarlanabilir.

Akış tipi üretim sistemleri dört ana prensip üzerine kurulmuştur:

i) **İş Akış Prensibi:** İş, malzeme ve ürünlerin düzgün ve düzenli bir şekilde üretim tesisleri boyunca akmaları gerekir.

ii) **İkame Edilebilen Parçaların Kullanımı:** Montaj hatları, büyük ölçüde, ikame edilebilen parçaların kullanımına bağlıdır. Diğer bir deyişle, üretimde kullanılan yarı mamul ve parçaların belirli spesifikasyonlara uygun olmaları gerekmektedir. Bu şekilde ana ürünü oluşturan parçaların, birbirinin yerine kullanılabilirmeleri mümkündür.

iii) **Minimum Uzaklık Prensibi:** İş akışının sürekliliğinin sağlanması ve kullanım alanının en iyi şekilde değerlendirilmesi, büyük ölçüde, kullanılan akış modelinin akılcı ve etken olmasına bağlıdır. Bu durumda üretim merkezlerinin birbirine yakın olmaları, hareketli bandlar ve benzeri malzeme aktarım ekipmanları ile birbirlerine bağlanmış olmaları gereklidir.

iv) **İş Yükünün Bölünmesi:** İş yükünün rasyonel bir şekilde bölünmesi veya montaj hatlarında iş gücünün bölünmesi, akış hattı üretim sisteminin önemli özelliklerinden biridir.

1.3. Seri (Akış Tipi) Ve Akış Hattı Üretim Sistemleri İçin Ön Şartlar

Seri üretim için en önemli şart, kitle taleptir. Akış hattı sistemleri, seri üretim sistemlerinin bir alt kolu olduğundan, kitle talep şartı bu sistemler için de geçerlidir.

"Kitle Talep" terimi, yalnız talep seviyesini değil, aynı zamanda talebin sürekliliğini de vurgular. Diğer bir deyişle, üretimin tümü pazar bulabiliyorsa ve bu durum süreklilik gösteriyorsa böyle bir sistemin kurulması anlam taşır. Bununla birlikte bazı durumlarda sürekli olmayan, ancak yüksek seviyelerdeki talebi karşılayabilmek için de akış hattı sistemleri kullanılabilir. Bu sistemlerin etkili bir şekilde işletilmeleri, aynı zamanda gerekli girdilerin (malzeme, yarı mamul vb.) istenilen yerde ve istenilen zamanda bulunmasına büyük ölçüde bağlıdır. Belirli bir aşamadaki girdi eksikliği kısa zamanda üretim hattının tamamen durmasına yol açacaktır. Sistemin etken olarak işleyebilmesi için hat dengesi ve güvenilirlik, en önemli faktörlerden ikisidir.

Bir ürünün akış hattı üzerinde verimli bir şekilde üretilmesi için, toplam iş yükünün, hattı meydana getiren iş istasyonları arasında oldukça eşit bir şekilde dağıtılması gerekir.

Akış hattı sistemlerinin önemli özelliklerinden biri de sistemdeki istasyonların birbirlerine bağımlı olmalarıdır. Diğer bir deyişle, belirli bir aşamadaki aksama, hızla diğer aşamaları da etkileyecektir. Bu durumda böyle sistemlerin alt sistemlerinin işletme açısından güvenilir olmaları zorunludur. Bu da, sistem içinde bakım planlamasının etkili bir şekilde uygulanmasını gerektirir.

1.4. Üretim Akış Hatları

Akış hatları, Transfer ve Montaj hatları olarak iki kısma ayrılırlar:

Transfer Hatları, transfer makinaları olarak da bilinirler; Büyük ve karmaşık makinalardan oluşan üretim sistemleridir. Bu hatlar düz (in-line) yada devirli (rotary) tiplerde olabilir. Ürün ya da malzemelerin bir yerden başka bir yere nakli, işgücü ile değil, otomatik hareket ve transfer yöntemiyle sağlanmaktadır. Böylece transfer hatları, transfer aletleriyle birbirine bağlanmış otomatik imalat makina serilerinin oluşturduğu üretim hatlarıdır.

Montaj Hatları: Bu hatların en belirgin özelliği, malzemelerin üretim hattı boyunca insan gücünden yararlanılarak transfer edilmeleri ve parça üzerindeki işlemlerin de yine bir hat boyunca sıralanmalarıdır.

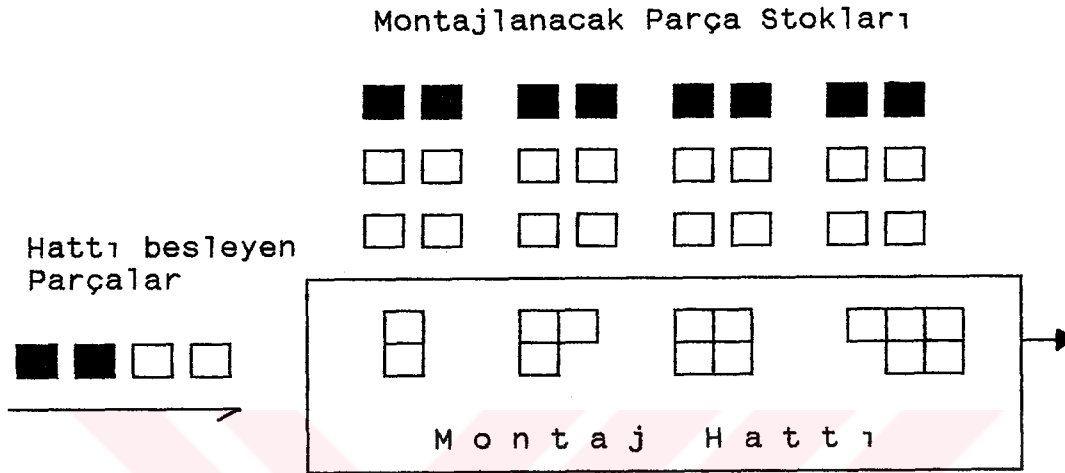
Akış hatlarının şu şekilde sınıflandırmasını yapmak da mümkündür:

i) Tek Modelli Hatlar: Bu hatlar, tek tip ürün ya da modelin üretiminde kullanılırlar.

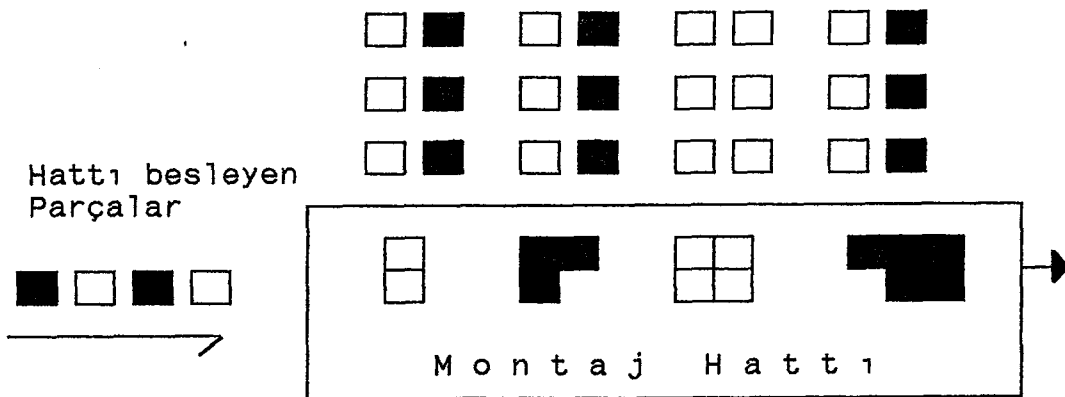
ii) Çok Modelli Hatlar: Bu hatlarda değişik modeller (ürünler) üretilir. Değişik modellerin üretimi ayrı ayrı kafieler halinde ve değişik zamanlarda yapılır. Belirli bir zamanda bir model (ürün) parti halinde üretilir, arkadan diğer modellerin üretimine geçilir. Modeller birbirlerine hiç bir zaman karıştırılmazlar(Bkz.Şekil 1.2). Modeller farklı ürünler ya da aynı ürünün farklı modelleri olabilirler. Her iki durumda da ürünler aynı olmayan fakat benzer üretim ihtiyaçları gösterirler. Pratikte montaj hattı birinci model için hazırlanır. Daha sonra ikinci, üçüncü vb. modellerin parti üretimi için hatta gerekli düzeltmeler yapılır.

iii) Karışık Modelli Hatlar: Aynı anda birden fazla benzer tipteki modellerin karışık olarak üretildiği hatlardır (Bkz. Şekil 1.3). Karışık modelli üretimin en önemli faydası, müşteri isteğini karşılamak üzere değişik modellerin sürekli olarak üretilmesi ve büyük bitmiş mamul stoklarını gerektirmemesidir. Modellerin değişik işlem zamanlarından doğan dezavantajlı yönleri ise; iş akışının düzenli olmaması, dolayısıyla daha fazla istasyon boş

zamanları, yarı bitmiş mamullerden oluşan yığınlardır (Acar ve Estaş, 1986).



Şekil 1.2 Çok Modelli Montaj Hattı (Dört istasyonlu)



Şekil 1.3 Karışık Modelli Montaj Hattı (Dört istasyonlu)

1.5. Tanımlar Ve Terminoloji

Montaj hatlarında bir mamulün montajı, birçok parça bileşen ve alt montajın biraraya getirilmesi ve bunların üzerinde birtakım işlemlerin uygulanmasıyla gerçekleştirilir. İşlemleri yapacak olanlar, hat boyunca sıralanmış olan işçi grupları veya diğer bir deyişle iş istasyonlarıdır. Bir montaj hattının temel özelliği, iş parçalarının bir istasyondan diğer bir istasyona hareket etmesidir.

Montaj hatları (dengelenmesi) ile ilgili karşımıza sık sık çıkacak olan terimler aşağıda kısaca açıklanmıştır:

İş İstasyonu: Üretim hattında yapılması öngörülen toplam iş miktarının bir kısmının yerine getirildiği yerdir. İstasyonlar, "Açık" ya da "Kapalı" olabilirler. Kapalı istasyonlarda iş, istasyon sınırları içinde yapılmak zorundadır. Açık istasyonlarda ise, operatör belirlenmiş sınırlara kadar istasyonun dışına hareket edebilir. Bu nedenle açık istasyonlardaki görevlerin tamamlanma zamanlarında esneklik söz konusudur (Dar-El, 1978).

İstasyon Zamanı: Bir iş istasyonuna atanan görevler için kullanılan gerçek zamandır. k istasyonunun istasyon zamanı ve toplam istasyon sayısı, sırasıyla S_k ve K ile gösterilir.

İş Elemanı veya Görev: Montaj hattında yerine getirilmesi gereken toplam iş yükünün rasyonel bir parçasıdır. Örneğin; otomobil üretiminde, belirli bir istasyonda, motordaki bir parçanın anahtarla sıkıştırılması, toplam iş yükünün daha fazla bölünemez rasyonel bir elemanıdır. Toplam iş elemanı sayısı N ile gösterilir.

İş Eleman (Görev) Zamanı: Bir görevin yapılabilmesi için gereken zamandır. i görevinin görev zamanı t_i ile gösterilir. Montaj hattı dengeleme problemlerini çözmek için geliştirilen tekniklerde bu zaman değerinin ya belirli bir sabit olduğu yada ortalaması μ_i ve varyansı σ_i^2 olan bir olasılık yoğunluk fonksiyonuna göre dağıldığı varsayılmıştır.

Çevrim Zamanı: Standart bir akış temposuyla bir üretim hattından tamamlanıp çıkan, ardarda iki ürün arasındaki farktır. Başka bir ifadeyle, bir iş istasyonuna, belirlenen görevlerin yapılabilmesi için ayrılan zamandır. Çevrim zamanı C ile gösterilir ve bu değer üzerinde aşağıda görülen alt ve üst sınırlar geçerlidir.

$$\max_{i=1,..,N} t_i \leq \max_{j=1,..,K} S_j \leq C \leq \frac{1}{D}$$

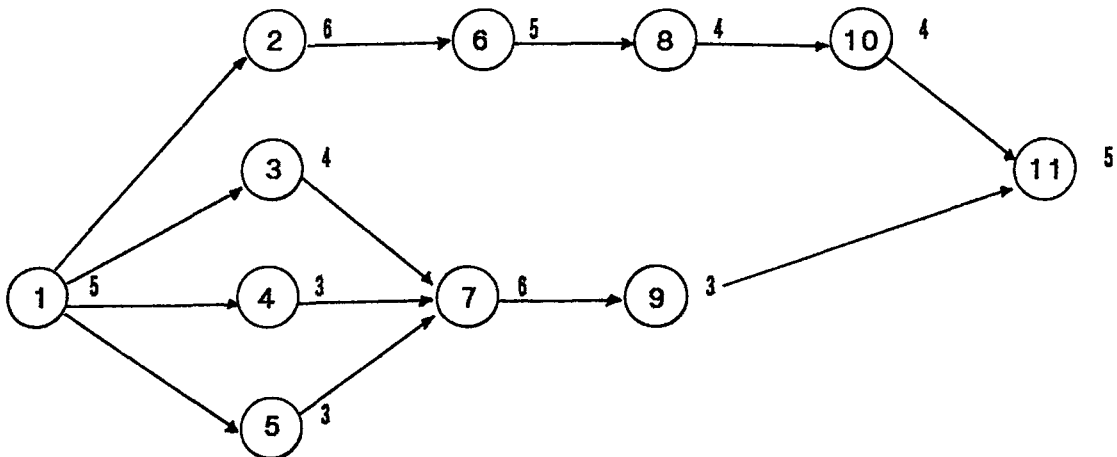
Boş Zaman: Çevrim zamanı ile istasyon zamanı arasındaki farktır. Bir hattın tasarım etkinliğinin bir

ölçüsü olarak bütün istasyonların boş zamanlarının toplamı alınabilir. Bu toplam , Toplam Boş Zaman olarak adlandırılır. Etkinliğin bir ölçüsü olarak Denge gecikmesi, toplam boş zamanın, ürünün hattın başından sonuna hareketi sırasında harcadığı toplam zamana oranıdır.

$$\text{Denge Gecikmesi} = \frac{100. [K.C - \sum_{i=1}^N t_i]}{K.C}$$

Tam dengelenmiş bir üretim hattında, denge gecikmesi sıfırdır.

Öncelik Diyagramı: Bir ürünün montajında yapılması gereken iş elemanlarının (görevlerin) işlem sıralarının grafiksel olarak gösterimidir. Şekil 1.4 'de 11 elemanlı bir hat için öncelik diyagramı verilmiştir. Dairelerin içindeki numaralar görevleri, dışındakiler ise görev zamanlarını ifade etmektedir.



Şekil 1.4 11 Elemanlı Bir Öncelik Diyagramı

Öncelik Matrisi: Öncelik matrisi bir üst üçgen matristir. Öncelik diyagramında , i görevini j görevi takip ediyorsa, i. satır ve j. sütun için giriş 1 olarak yazılır. Diğer durumlarda ise, bu değer sıfırdır. Şekil 1.4 'de gösterilen öncelik diyagramı için oluşturulan öncelik matrisi Şekil 1.5 'de verilmiştir.

		Görevler										
		1	2	3	4	5	6	7	8	9	10	11
G ö r e v l e r	1	-	1	1	1	1	1	1	1	1	1	1
	2		-	0	0	0	1	0	1	0	1	1
	3			-	0	0	0	1	0	1	0	1
	4				-	0	0	1	0	1	0	1
	5					-	0	1	0	1	0	1
	6						-	0	1	0	1	1
	7							-	0	1	0	1
	8								-	0	1	1
	9									-	0	1
	10										-	1
	11											-

Şekil 1.5 (Şekil 1.4) deki Öncelik Diyagramına ait
Öncelik Matrisi

Bir montaj işleminin öncelik yapısı, Esneklik Oranı (E-Oranı) ile belirlenebilir. E-Oranı, N elemanlı bir montaj işleminden elde edilebilen uygun sıraların sayısının bir ölçüsüdür.

$$E-Oranı = \frac{2.Y}{N.(N-1)}$$

Y : Öncelik matrisindeki sıfırların sayısıdır.

Öncelik diyagramı seri şekilde sıralıysa, bu oran sıfır'a, öncelik ilişkileri olmayan diyagramlar için ise bu oran bir'e eşittir. Şekil 1.4 'deki öncelik diyagramının E-Oranı 0.418 dir. E-Oranının sıfıra yakın olması, esnekliğin az olduğunu göstermektedir.

Görevlerin Paralelliği: Bir görevin, birden fazla istasyonda yapılmasına mücade edilmesidir.

İstasyonların Paralelliği: Hat boyunca belirli noktalarda, denk iş istasyonlarına mücade edilmesidir. Paralellik kavramı, bir hat tasarımını çeşitli şekilde düzenlemek için kullanılır; Böylece, denge etkinliği geliştirilebilir ve çevrim zamanını geçen iş elemanları da rahatlatılabilir.

Bölgeleme Kısıtları: Bu kısıt, iki şekilde ifade edilebilir: Bunlardan ilki pozitif bölgeleme kısıtlarıdır ki, burada belirli görevlerin aynı yerde veya aynı istasyonda yapılması gerekliliği vardır. İkincisi ise negatif bölgeleme kısıtlarıdır; Bu kısıtta ise, görevlerin birbirlerinden uzaklaştırılması, başka bir ifadeyle, bu görevlerin aynı istasyonda yapılmaması durumu söz konusudur.

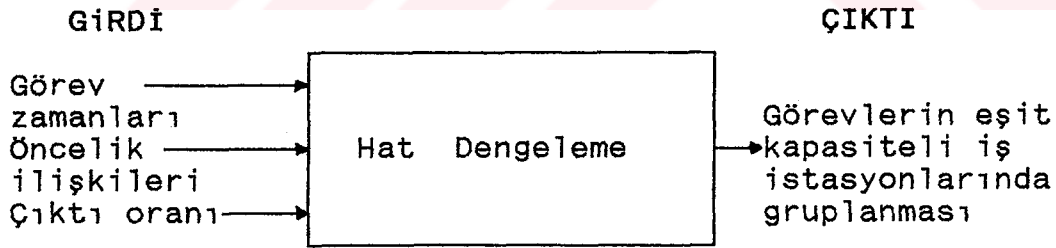
İstasyonun Tıkanması (Blocking) ve Açlığı

(Starving): Eğer istasyonda tamamlanmış birimler, yetersiz alan yüzünden stoklanamıyorsa, tamamlanmış

birimler istasyonda kalmak zorundadırlar ki, bu istasyon tıkanmıştır. Böylece bu istasyon, stoklamada herhangi bir alan elverişli hale gelene kadar boş bekleyecektir. Eğer işlemi başlatmak için, kullanmak üzere hiçbir birim yok ise, istasyon açtır (starved) denilir (Erel,1987).

1.6. Montaj Hattı Dengeleme Problemi Ve Sınıflandırılması

Montaj Hattı Dengeleme Problemi (MHDP) şu şekilde tanımlanabilir: Montaj işleminin yapılabilmesi için gerekli görevler, bu görevlerin aldıkları süreler ve aralarındaki öncelik ilişkileri verildiğinde, bir performans ölçüsü eniyilenecek şekilde, görevlerin sıralı istasyonlara atanmasıdır. Bir Montaj hattı sistemi şekil 1.6 'da gösterilmiştir.



Şekil 1.6 Montaj Hattı Dengeleme Sistemi
(Dervitsiotis,1981).

Kullanılan performans ölçülerinin önemli bir bölümü, iki ana grupta toplanabilir: Birinci grup, hat boyunca yerleştirilen istasyon sayısı, ikinci grup ise istasyonlara verilen sürelerdir(çevrim zamanı). Birinci grup insangücü

maliyetini enküçükmekte, ikinci grup ise üretim miktarını enbüyüklemektedir. Problemin formüle edilmesinin kolaylığına rağmen, yukarıdaki performans ölçülerini eniyileyen iş gruplarının belirlenip istasyonlara atanması, diğer bir deyişle problemin çözümü oldukça güçtür (Erel, 1991). N tane görev ve bu görevler arasında r tane öncelik ilişkisi olan bir problem için yaklaşık olarak $N!/2^r$ değişik hat tasarımı elde edilebilir (Ignall, 1965). Endüstrideki uygulamalar gözönüne alındığında bu rakam, günümüz bilgisayar kapasitelerinin çok üzerindedir.

Problemin çözümü için geliştirilen yöntemler, en iyi çözümü bulan ve sezgisel yöntemler olmak üzere ikiye ayrılırlar; En iyi çözümü bulan yöntemler, çok uzun bilgi işlem zamanına ihtiyaç duyduklarından, genellikle akademik araştırmalar olarak kalan ve pratikte fazla uygulama imkanı bulunmayan yöntemlerdir. Sezgisel yöntemler ise, problemin bir çözümünü oldukça hızlı elde eden yöntemlerdir.

Montaj Hattı Dengeleme Problemlerinde, çok genel iki farklı durum sözkonusudur:

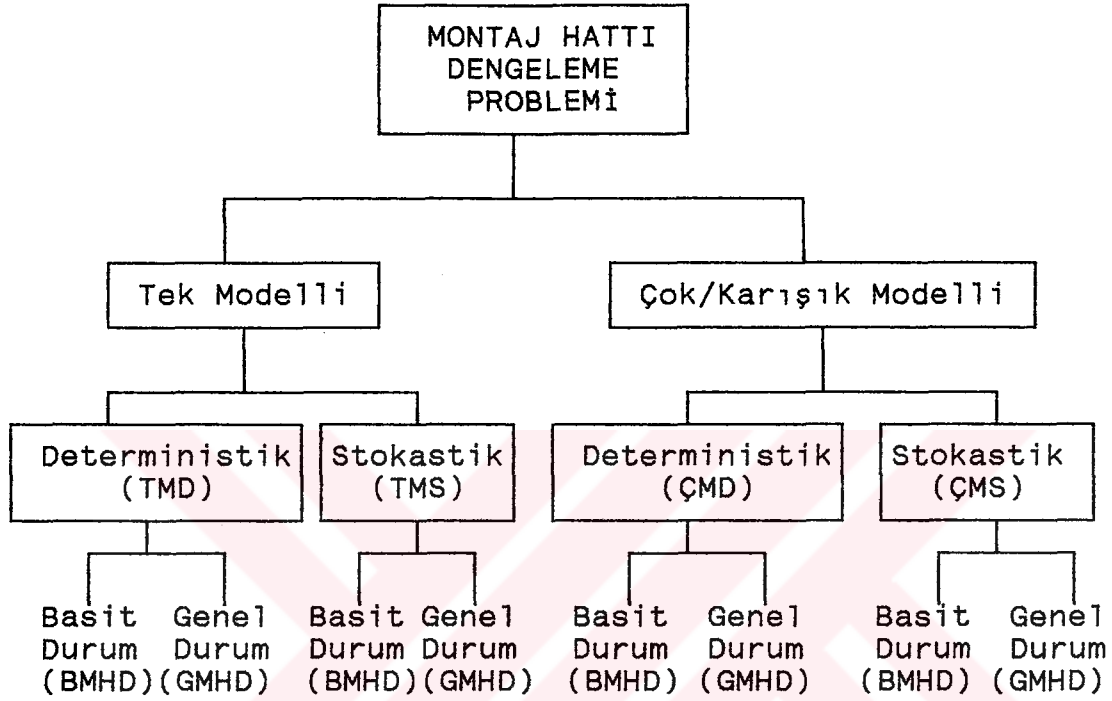
- 1- İşlem zamanlarının belirli olması (Deterministik MHD)
- 2- İşlem zamanlarının değişken olması (Stokastik MHD)

Deterministik MHD problemlerinde, görev zamanlarının verilmiş olduğu ve bu zamanların bir birimden diğerine herhangi bir değişim göstermediği varsayılmaktadır. Bu varsayım özellikle robot teknolojisinin uygulama alanı bulabildiği, ileri teknoloji sanayilerinde geçerlidir.

Stokastik MHD problemlerinde ise, görev zamanları, belirli bir dağılımla ifade edilir. İnsan unsuru, görev zamanlarının değişken olmasına yol açmaktadır. Söz konusu değişkenliğin sebepleri arasında yorulma, dikkatin dağılması, yetersiz nitelikli işgücü, iş tatminsizliği, hatalı girdiler, araç/gereç bozulmaları sayılabilir. Bu durum istasyonlara atanan işlerin aldıkları toplam zamanın, çevrim zamanını aşmasına ve dolayısıyla da bazı görevlerin bitirilememesine sebep olabilmektedir. Özellikle işler arasındaki öncelik ilişkileri gözönüne alındığında, bazı görevlere hiç başlanamamaktadır. Bu tür dengeleme problemlerinde araştırmacıların çoğu, görev zamanlarının Normal dağılıma göre değer aldıklarını, bazıları ise görev zamanlarının değişkenlik katsayılarının (s/μ) tüm işler için sabit olduğunu varsaymışlardır (Erel, 1991).

Montaj Hattı Dengeleme Problemleri, 4 grupta toplanabilir: Tek Modelli Deterministik (TMD), Tek Modelli Stokastik (TMS), Çok/Karışık Modelli Deterministik (ÇMD), ve Çok/Karışık Modelli Stokastik (ÇMS) problemler.

Şekil 1.7 'da MHD problemlerinin sınıflandırılması verilmiştir.



Şekil 1.7 Montaj Hattı Dengeleme Problemlerinin Sınıflandırması

BMHD, montaj hattı dengeleme problemlerinin en basit ve orjinal şeklidir. Eğer modelde bir takım kısıtlamalar veya faktörler (Paralel istasyonlar, bölgeleme kısıtları vb.) dahil edilirse, problem Genel Hat Dengeleme (GMHD) problemine dönüşür (Ghosh and Gagnon, 1989).

1.6.1. Tek modellenli deterministik (TMD) hat dengeleme problemi

Bu tür problemlerde, üretim hattında bir tek modelin (ürünün) üretilmesi sözkonusudur ve görev zamanları sabittir. Buna göre görevler, öncelik ilişkileri sağlanacak ve belirli bir performans ölçüsü eniyilenecek şekilde iş istasyonlarına atanırlar. Bir istasyona atanan görevlerin zamanlarının toplamı, çevrim zamanına eşit ya da ondan küçük olmalıdır.

Problemin varsayımları aşağıda verilmiştir:

- i) Üretim hattında, tek bir ürünün montajı yapılmaktadır.
- ii) Görev zamanları bilinen sabitlerdir.
- iii) Problemin öncelik diyagramı bilinmektedir.
- iv) Bir görevin iki ya da daha fazla istasyon arasında bölüştürülmesine izin verilmemektedir.
- v) Herbir görev, kendisinden önce gelen görevler tamamlanmadan başlayamaz.

Bu tür montaj hattı dengeleme problemlerinin çözümü için çeşitli performans ölçüleri kullanılmıştır. Genellikle kullanılan performans ölçüsü, istasyonların sayısının enküçüklenmesidir. Toplam boş zamanın enküçüklenmesi, verilen bir istasyon sayısı için çevrim zamanının

enküçüklenmesi, ve denge gecikmesinin enküçüklenmesi, literatürde bahsedilen diğer önemli performans ölçüleridir.

Toplam boş zamanın enküçüklenmesi için amaç fonksiyonu aşağıdaki gibi ifade edilebilir:

$$\text{Min Toplam Bos Zaman} = \text{Min } Z = K.C - \underbrace{\sum_{i=1}^N t_i}_{\text{sabit}}$$

Yukarıdaki formülde yer alan sabit kısım dikkate alınmaz ise, montaj hattının dengelenmesi, aşağıdaki yollardan birisi izlenerek gerçekleştirilir.

- i) Min $Z = K$, C veriliyor
- ii) Min $Z = C$, K veriliyor.

Çoğu durumda, yönetim üretim oranını belirlediğinde, çevrim zamanı da belli olacağına göre, problemi en az istasyon sayısının bulunması şeklinde indirgeyebiliriz.

Problemin kısıtları aşağıda verilmiştir;

- i) Bütün görevler yapılmak zorundadır.
- ii) Hiç bir görev birden fazla atanamaz.
- iii) Herhangi bir istasyondaki işyükü, çevrim zamanını geçemez.
- iv) Eğer öncelik diyagramında x görevi y görevinden

önce geliyorsa, o zaman x görevi herhangi bir istasyona atanmadan y görevi yapılamaz.

Problemin formüle edilmesi, oldukça kolay olmasına rağmen en az istasyon sayısını bulmak için uygun görev sıralarının belirlenmesi, çok büyük bir çaba gerektirmektedir (Erel, 1987).

1.6.2. Tek modellenli stokastik (TMS) hat dengeleme problemi

Stokastik hat dengeleme problemi, şu şekilde tanımlanabilir: Sonlu bir görev seti verilir ve bu görevlerden herbiri bir olasılık dağılımına göre dağılmış görev zamanlarına sahiptir. Buna göre problem, öncelik ilişkileri dikkate alınarak ve belirli bir performans ölçüsü eniyilenecek şekilde görevlerin uygun iş istasyonlarına atanmasıdır.

Görev zamanları stokastik olduğunda, görev aşağıdaki iki sebepten dolayı tamamlanamamaktadır.

- 1-Ya görev, çevrim zamanı içinde tamamlanamaz, ya da
- 2-Görev, öncelik diyagramında tamamlanmamış bir görevin takipçisidir.

Bu problem türünde genellikle, toplam sistem maliyetinin, işgücü ve tamamlanmama (incompletion) maliyetinden (Birimlerin hatta ilerlemeleri esnasında,

tamamlanamayan görevlerin sebebiyet verdiği maliyet) oluştuğu söylenebilir.

Problemde, görev zamanları, parametreleri belli bilinen bir olasılık dağılım fonksiyonuna göre dağılırlar. Hatta tamamlanamayan görevler, ya en kısa sürede tamamlanmak üzere hattın dışına alınırlar ve orada tamamlanırlar, ya da bitmemiş ürün olarak ayrılırlar.

TMD Hat dengeleme problemindeki (i). varsayım hariç diğer varsayımlar, bu problem için de geçerlidir.

Literatürde Stokastik MHD problem için en fazla rastlanılan amaç, toplam işgücü ve toplam beklenen tamamlanmama maliyetlerinden oluşan toplam sistem maliyetinin enküçüklenmesi şeklindedir.

$$\text{Min } Z = \text{Toplam Sistem Maliyeti} = C \times L \times K + \sum_{\text{Bütün } G \text{ ler}} I(G) \cdot Pr(G)$$

Eşitlikteki L, birim işgücü maliyeti, G tamamlanmayan kombinasyonların seti, I(G), G kombinasyonun tamamlanmama maliyeti, Pr(G) ise G kombinasyonunun meydana gelme olasılığıdır. Yukarıdaki formülasyonda görülen ilk terim, birim başına işgücü maliyetini, ikinci terim ise, birim başına beklenen tamamlanmama maliyetini ifade etmektedir.

Bu tür problemlerde, uygun sıraların değerlendirilmesi, deterministik durumdan daha karmaşıktır (Kottas and Lau, 1976).

1.6.3. Çok/Karışık modellenli deterministik (ÇMD) hat dengeleme problemi

Tüketici sayılarının ve bilinç düzeylerinin artması ile diğer imalatçı kuruluşların rekabeti, tek tip ürünün seri üretimini engellemektedir. Bu durumda çok modellenli ve karışık modellenli hat tasarımları gündeme gelmektedir. Bu tür problemlerde, görev zamanları deterministiktir ve birden fazla modelin aynı montaj hattında üretilmesi sözkonusudur.

Çok modellenli hatlarda ya farklı ürünler ya da aynı ürünün değişik modelleri üretilir. Her iki durumda da benzer üretim süreçleri sözkonusudur. Uygulamada montaj hattı belirli bir modelin üretilmesi için hazırlanır, o modelin üretimi bittikten sonra, ikinci modelin üretimi için hazırlık yapılır. Bu durumda her model üretimi için hat dengeleme problemleri, birbirinden bağımsız olarak tek model montaj hattı dengeleme problemleriyle çözümlenir. Bunun yanısıra bu sistemlerde bir de model sıralama probleminden bahsedilir.

Model sıralama probleminde, tezgah hazırlama maliyetlerini enazlayan sıralamanın seçilmesi için

geliştirilmiş yöntemlerden faydalanılır (Dal ve sınır, atama metodu vb.). Özetle, çok modellenli üretim hatlarının tasarımı, aşağıda belirtilen ana sorunların çözümünü içerir.

- Her farklı model (kafile) için hat nasıl dengelenecek ?
- Kullanılacak kafile büyüklükleri ne olacak ?
- Modeller nasıl bir sıralama ile üretilecekler ?

Karışık modellenli hatlarda, bir ürünün değişik modelleri aynı anda ve karışık olarak üretilirler. Oldukça karmaşık tasarım ve işletme problemleriyle karşılaşılan bu üretim hatları için şu ana kadar geliştirilen yöntemlerin sayısı yeterli değildir. Genelde bu hatların tasarımı aşamasında karşılaşılan ana problemler şöyle özetlenebilir:

- Modeller hangi sırayla hatta yüklenecekler ?
- Hat nasıl dengelenecek ?

Karışık modellenli hatlarda, kuramsal olarak büyük miktarlarda bitmiş ürün stoklarına ihtiyaç olmayıp, çok modellenli hatların tersine tüketici istekleri, sürekli bir şekilde yapılan üretimle karşılanmaktadır.

Karışık modellenli hatların dengelenmesi için kullanılan bazı sezgisel algoritmalarda görevlerin operatörlere

atanmasında, tek modelli hatlarda olduğu gibi çevrim zamanına göreden ziyade, genellikle günlük veya bütün bir çalışma devresinin esas alındığını görmekteyiz (Acar ve Estaş, 1986).

Problem varsayımları (i). varsayım hariç TMD problemin varsayımlarıyla aynıdır. Ancak aşağıdaki varsayımın ilave edilmesi gereklidir.

i) Aynı ürünün çeşitli modelleri aynı hatta işlenmelidir.

Karışık modelli montaj hattı dengeleme problemlerinde öncelik ilişkilerini belirleyen öncelik diyagramı, tek model montaj hattı dengeleme problemlerindeki öncelik diyagramından oldukça farklıdır.

Thomopoulos (1970), iki ya da daha fazla modelin öncelik diyagramlarını tek bir diyagram halinde birleştirerek, birleştirilmiş (Combined) öncelik diyagram kavramını geliştirmiştir. Bunu biçimsel şekilde Macaskill (1972) formüle etmiştir.

Karışık modelli montaj hattı dengeleme problemlerinde öncelik diyagramlarının formülasyonu ve ifadesi şöyledir:

G_m , m modeline ait öncelik diyagramıdır. G_m , $N(m)$

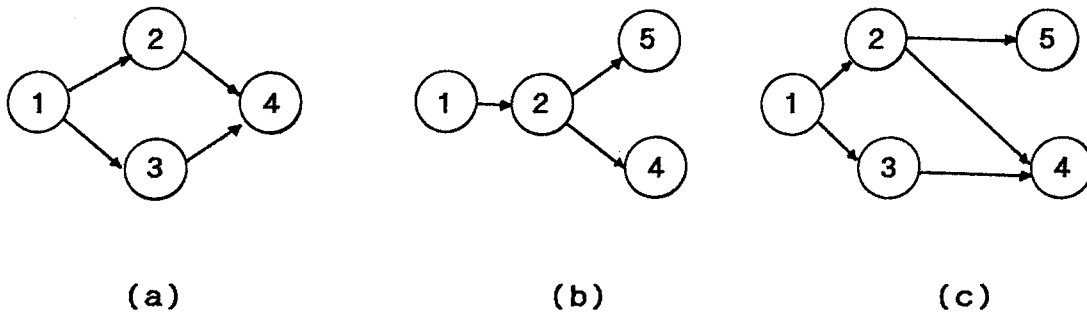
düğüm ve $L(m)$ ok setinden oluşur.

$$N(m) = [n(m)_1, n(m)_2, \dots] \quad m=1, \dots, P$$

$$L(m) = [l(m)_1, l(m)_2, \dots] \quad i=1, \dots, N$$

Düğüm, görev, oklar ise öncelik ilişkilerini gösterir. Eğer bir ok $l(m)_i$, $n(m)_i$ başlangıç düğümü ve $n(m)_j$ terminal düğümüne sahipse, o zaman $n(m)_i$ ile ifade edilen görev, $n(m)_j$ görevi başlamadan önce tamamlanmak zorundadır. $t(m)_i$ ve $t(m)_j$ terimleri de sırasıyla $n(m)_i$ ve $n(m)_j$ nin görev zamanlarını göstermektedir.

Şekil 1.8 'de, 2 ayrı modelin öncelik diyagramları ve bunların tek bir diyagram halinde birleştirilmiş durumu gösterilmektedir:



Şekil 1.8 1.Ürün (a) ve 2. Ürün (b) 'e ait Öncelik Diyagramları ve Birleştirilmiş Öncelik Diyagramı (c)

$M=\{1,2,...,P\}$ modeller seti için öncelik diyagramları, $G_1, G_2, G_3, ..., G_P$ graflarıyla (serimleriyle) ifade edilir. Bu öncelikler $N(M)$ düğümlü ve $L(M)$ oklu G_M serimiyle ifade edilebilir.

$$N(M) = N(1) \cup N(2) \cup ... \cup N(P)$$

$$L(M) = L(1) \cup L(2) \cup ... \cup L(P)$$

G_M , Birleştirilmiş öncelik diyagramı olarak adlandırılır. Eğer oklar, $n(m)_i$ den $n(m)_j$ ye bir zincir oluşturuyorsa, $(n(m)_i, n(m)_j)$ oku gereksizdir (fazladır). Bu fazladan okun ihmal edilmesi gereklidir. Eğer G_M deki $n(M)_i$, G_M deki $n(m)_j$ ye karşılık geliyorsa, o zaman $n(M)_i$ nin zamanı $t(M)_i$, $t(m)_j$ ye eşittir.

Birleştirilmiş diyagram G_M , farklı modeller arasındaki öncelik ilişkilerinde bir çelişkinin olmadığı durumlar için uygundur. Örnek verecek olursak, eğer 1. modelde A görevi B görevinden önce geliyorsa, diğer hiçbir modelde B görevi A görevinden önce gelemez (Macaskill, 1972).

1.6.4. Çok/Karışık modellenli stokastik (ÇMS) hat dengeleme problemi

ÇMS hat dengeleme probleminde, görev zamanlarının stokastik olmasına izin verilir. Görev zamanları TMS'de olduğu gibi parametreleri bilinen dağılımlarla ifade edilirler. ÇMD' de ifade edilen Birleştirilmiş öncelik

diyagramı bu problem için de geçerlidir.

ÇMS hat dengelemede problemi zorlaştıran bir çok faktör vardır. Bunları, öğrenme etkileri, işçi hüner seviyesi, iş tasarımı ve işçinin iş zamanı değişkenliği vs. şeklinde sıralayabiliriz. Bunlardan dolayı, birimlerin tamamlanmadan hat boyunca ilerlemeleri ve dolayısıyla da tamamlanmayan birimlerin hattın dışında tamamlanması, birtakım maliyetlere sebep olmaktadır (Ghosh and Gagnon, 1989).

Oldukça karmaşık olan bu tür dengeleme problemleriyle ilgili çok az sayıda çalışma bulunmaktadır.

1.7. Çalışmanın Amacı

Montaj Hattı Dengeleme Problemi, Endüstri Mühendisliği ve Yöneylem Araştırmasında, üzerinde en çok çalışılan konulardan birisi olmasına rağmen, MHD probleminin bir alt sınıfı olan Karışık MHD problemleri için yapılan çalışmalar oldukça azdır. Halbuki tüketici sayılarının ve bilinç düzeylerinin artması ile diğer imalatçı kuruluşların rekabeti, tek tip ürünün seri üretimini engellemekte olup, çok ve karışık modelli hat tasarımlarını gündeme getirmektedir.

Bu çalışmada, Karışık modelli MHD problemlerinin çözüm

yöntemleriyle ilgilenilmiştir.

İlk olarak Karışık Modelli MHD problemleri için geliştirilen optimal teknikler incelenmiş ve pratik için gerekli birtakım varsayımların ilavesiyle, daha iyi performansın sağlandığı alternatif optimal metotlar geliştirilmiştir.

MHD Problemi, kombinatoryal türden bir problem olduğu için, geliştirilen bu optimal metotlar ne kadar etkili olsa da, çözüm, belirli problem boyutuyla sınırlı olacağından, aynı tür problemin çözümü için bir de sezgisel prosedür sunulmuştur.

Son olarak, bu tür problemler için, şu ana kadar geliştirilen metotların hepsinde sadece bir tek amacın sağlanmasına yönelik çalışmaların yapılması ve birden fazla amacı dikkate alarak yönetime karar verme esnekliği tanıyan çalışmaların olmamasından dolayı, Çok amaçlı karar verme tekniklerinden Amaç Programlama tekniği kullanılarak bir 0-1 Amaç Programlama modeli geliştirilmiştir.

BÖLÜM 2

KAYNAK ARAŞTIRMASI

Bu bölümde, literatürdeki teknikler gözden geçirilmiş ve herbir hat dengeleme problemiyle ilgili kaynak araştırmasının ayrı ayrı yapılması şeklinde bir yol izlenmiştir.

2.1. Tek Modelli Deterministik Hat Dengeleme Problemi(TMD)

TMD Montaj Hattı Dengeleme Problemi, en fazla araştırılan ve üzerinde en fazla makalenin yayınlandığı problemlerdir.

Bu problemle ilgili ilk çalışma Salveson'a (1955) aittir. Salveson (1955) MHD problemini doğrusal tamsayılı programlamayla modellemiştir. Modelinin amaç fonksiyonu, istasyonlardaki toplam boş zamanın enküçülenmesidir. İstasyonlara görev atamalarının tüm mümkün kombinasyonlarının bir araya getirilmesi, modelin pratikten ziyade akademik seviyede olduğunu göstermektedir.

Bowman (1960) problemi, doğrusal 0-1 tamsayılı programlama şeklinde formüle ederek, görevlerin bölünmesi problemini ortadan kaldırmıştır.

Thangavelu ve Shetty (1971), Bowman ve White'in 0-1

tamsayılı programlama formülasyonunun bir geliştirilmiş versiyonunu sunmuşlar ve problemin çözümü için Balas additive algoritmasını kullanmışlardır.

Talbot ve Patterson (1984) tek model montaj hattı dengeleme problemleri için bir tamsayılı programlama algoritması geliştirmişlerdir. Çözüm prosedürü, tüm mümkün görev atamalarının sistematik birerlemesiyle ilgilidir. Prosedür, 50 ya da daha az görevden oluşan hatlar için optimal dengeleri kabul edilebilir bir zamanda bulabilmektedir.

Patterson ve Albracht (1975), literatürdeki diğer formülasyonlardan daha etkili bir tamsayılı programlama formülasyonu sunmuşlardır. Ayrıca formülasyonun çözüm prosedürü için de bir arama (search) algoritması geliştirmişlerdir.

Pinto, Dannenbring ve Khumawala (1975), görevlerin paralel olduğu durumları içeren dengeleme problemleri için bir tamsayılı programlama modeli önermişler; çözüm için de dal ve sınır algoritmasını kullanmışlardır.

Klein (1963), problemi bir en kısa yol problemi olarak modellemiştir. Metot, görevlerin mümkün tüm sıralarının setini oluşturduğundan, büyük ölçekli problemler için elverişli değildir.

Gutjahr ve Nemhauser (1964), bir sonlu yönlü serimde en kısa yolun bulunmasıyla ilgili bir algoritma önermişlerdir. Algoritmanın ana zorluğu, oluşturulan düğüm sayısının, problemin boyutuyla üstel olarak artmasıdır. Metot, Klein'in (1963) en kısa yol yaklaşımı üzerine anlamlı katkılar sağlamıştır; Zira metot, bütün uygun durumları üretmemektedir.

Jackson (1956), Held ve arkadaşları (1963), Kao ve Queyranne (1982), Held ve Karp (1962) ve Schrage ve Baker (1978), Dinamik programlama metoduyla montaj hatlarının dengelenmesi üzerine çalışmalar yapmışlardır. Kao ve Queyranne'nin (1982) formülasyonu, Schrage ve Baker'ın (1978) formülasyonunun bir değişik şeklidir. Dinamik programlama yaklaşımları, hesaplama olarak kolay kontrol edilemezken, Schrage ve Baker (1978) ve Kao ve Queyranne (1982) tarafından kullanılan etiketleme şemaları, geleneksel dinamik programlama algoritmalarına (örnek Held ve Karp (1962)) kıyasla hem zaman hem de bellek bakımından oldukça etkilidir.

Hu (1961), Van Assche ve Herroelen (1979), Johnson (1981) ve Wee ve Magazine (1981), basit montaj hattı problemlerinin çözümü için genel tamsayılı programlamaya bağlı olmayan özel dal ve sınır yaklaşımları kullanmışlardır.

Gunter ve Arkadařları (1983), genel MHD problemleri iin bir ama programlama modeli geliřtirmişlerdir.

Deckro (1989), bir montaj hattı dengeleme probleminde evrim zamanı ve istasyon sayısını aynı anda dikkate alarak, tek amaçlı bir maliyet modeli nermiştir. Bu model, Patterson ve Albracht (1975) ile Thangavelu ve Shetty'nin (1971) bilinen 0-1 tamsayılı formölasyonundan geliştirilmiştir.

Deckro ve Rangachari (1990), dengeleme problemi iin bir 0-1 tamsayılı ama programlama modeli geliřtirmişlerdir. Model, bölgeleme, sıralama, boş zaman, evrim zamanı vb. gibi eřitli operasyonel ihtiyaları aynı anda dikkate alarak hat dengelerinin esnekliėinin artırılmasını saėlamıştır. Modelde, Patterson ve Albracht'ın (1975) 0-1 tamsayılı programlama formölasyonundan faydalanılmıştır.

Malakooti (1991) MHD problemi iin ok amaçlı karar verme yaklařımını kullanmıştır. Ama fonksiyonları; evrim zamanının, istasyon sayısının ve operasyon maliyetinin tümünün enküüklenmesidir. Problemin özümü iin bir etkileřimli (interactive) algoritma kullanılmıştır.

Shin ve Hokey (1991), dengeleme problemleri iin yeni bir yaklařımdan söz etmektedirler. alıřma, hem evrim

zamanındaki ve hem de istasyon sayısındaki deęişkenlięi göz önüne alarak, geleneksel (bilinen) MHD yaklaşımlarını genelleştirmektedir. Bu yeni yaklaşımın en önemli farklılıęı, işgücü verimlilięini enbüyüklemek (ya da boş zamanı enküçükmek) amacıyla, çevrim zamanı ve istasyonların sayısının karar deęişkenler olarak göz önüne alınmasıdır.

Tek modellenli deterministik MHD problemleri için, yukarıdaki optimal metotlara ilave olarak bir de geliştirilen sezgisel metotlar vardır. Bunları aşağıdaki şekilde sıralayabiliriz:

- . Helgeson ve Birnie'nin (1961) Sıralanmış Pozisyonlu Ağırlık Metodu (RPWT).
- . Tonge'nin (1960) üç safhadan oluşan sezgisel prosedürü
- . Kılbridge ve Wester'in (1961) sezgisel metodu.
- . Hoffman'ın (1963) öncelik matrisi metodu.
- . Arcus'un (1966) rasgele uygun sıraları üreten COMSOAL Metodu.
- . Nevins'in (1972) genel amaçlı sezgiseli.
- . Dar-El'in (1973) büyük ölçekli tek model montaj sistemleri için geliştirdięi MALB metodu.
- . Pinto, Dannenbring ve Khumawala'nın (1978), Gutjahr ve Nemhauser'in (1964) en kısa yol algoritmasını esas alan Sezgisel Serim metodu.

- . Dar-El ve Rubinovitch'in (1979) MUST metodu.
- . El-Sayed ve Raouf Tsui'nin (1980) Kritik Yol sezgisel metodu.
- . Akagi, Osaki ve Kikuchi'nin (1983) iki safhadan oluşan Paralel Atama metodu (PAM).
- . Agrawal'ın (1985) En Büyük Küme Kuralı sezgiseli.
- . Dar-El ve Rubinovitch'in (1991) genellikle küçük miktarlar gerektiren ve görev zamanları oldukça uzun yeni ürünlerin montajı için geliştirdikleri Hat dengeleme modeli. (Bu model, endüstriyel öğrenme nedeniyle performanstaki değişmeyi de dikkate almaktadır.)

Yukarıdaki metotlara ilaveten, Bhattacharjee ve Suhu (1990), hem basit hem de genel MHD problemlerinin kompleksitelerini test etmek için problem üretilmesini sağlayan bir metot sunmuşlardır.

2.2. Tek Modelli Stokastik Hat Dengeleme Problemi (TMS)

Deterministik görev zamanlı MHD problemlerini çözen bir çok tekniğin geliştirilmesine rağmen, stokastik görev zamanlı problemler için yapılan çalışmalar, göreceli olarak o kadar fazla değildir. Problemin matematiksel yapısındaki karmaşıklık, bu konuda fazla çalışma yapılamamasının en önemli nedenlerinden biridir.

Moodie ve Young (1965) bu konuda, ilk sezgisel metot geliştirenler arasındadır. Görev zamanları sabit ve Normal dağılmış montaj hatlarının dengelenmesi için geliştirdikleri sezgisel prosedür iki safhadan oluşmaktadır. İlk safhada, "En büyük görev zamanı" kuralına göre ilk denge oluşturulmakta, ikinci safhada ise görevler kaydırılarak, istasyonlardaki boş zamanın yok edilmesine çalışılmaktadır.

Mansoor ve Ben-Tuvia (1966), normal dağılmış görev zamanlarıyla birlikte tam dengelenmiş istasyon sayıları verildiğinde, en iyi çevrim zamanının belirlenmesi üzerine çalışmışlardır. Metot, iki plan çerçevesinde toplam iş gücü maliyetlerini enküçükleyen optimal bir çevrim zamanını araştırmaktadır. İlk plan sabit teşvik oranlarına bağlıdır, ikinci plan ise işçinin kazanç için sarfettiği çabayla ilişkilidir. Bu araştırma, ekonomik amaç kriterinin kullanımı açısından oldukça önemlidir.

Reeve ve Thomas (1973), stokastik montaj hatları için 4 çözüm prosedürünü karşılaştırmışlardır. Bunlar; T&T (Trade ve Transfer), Dal ve sınır, Sezgisel dal ve sınır, ve T&T ile sezgisel dal ve sınır'ın bir kombinasyonu olan BABTAB metotlarıdır. Reeve ve Thomas (1973) başlangıç bir denge ve çevrim zamanı verildiğinde, görev zamanlarının normal dağıldığı varsayımı altında bir ya da daha fazla istasyonun, çevrim zamanını geçme olasılığını enküçükmeye

çalışmışlardır.

Kao (1976, 1979) ve Sniedovich (1981), problem için bir dinamik programlama modeli önermişlerdir. Amaç, bütün öncelik ilişkileri sağlanacak ve istasyon sayısı enküçüklenecek şekilde görevlerin istasyonlarda gruplanmasıdır. Model her bir istasyondaki iş yükü olasılığının, önceden belirlenmiş bir değerle sınırlı çevrim zamanından büyük olmamasını sağlar. Görev zamanları; Poisson, gamma, binomial veya negatif binomial'e yaklaştığında, prosedür optimal çözüm vermektedir.

Sphicas ve Silverman (1976), görev zaman dağılımlarının belirli sınıfları için, MHD problemlerinin deterministik ve stokastik formülasyonlarının, birbirine denk olacak şekilde dönüştürülebildiklerini göstermişlerdir.

Kottas ve Lau (1973). çevrim zamanının verildiği bir montaj hattı için, tamamlanmama ve iş gücünün toplam maliyetini enküçükleyen bir sezgisel metod geliştirmişlerdir. Ayrıca Kottas ve Lau (1976), hattın beklenen tamamlanmama maliyetinin değerlendirilmesi için de bir etkin prosedür tanımlamışlardır. Prosedür tamamlanmamış görevlerin tüm mümkün kombinasyonlarını belirlemekte ve bu kombinasyonların maliyet ve olasılıklarını değerlendirmektedir. Yine Kottas ve Lau (1981), çeşitli

uygun hat tasarımları üreten sezgisel bir prosedür geliştirmişlerdir.

Shtub (1984), istasyonlarda birden fazla insanın çalıştığı ve görev zamanlarının stokastik olduğu montaj hatlarının tasarımı için bir sezgisel prosedür geliştirmiştir. Prosedür, Kottas ve Lau'nun (1973) metoduna oldukça benzemektedir.

Sculli (1979), sistemin başlangıç tasarımından sonra, prosesteki çeşitli dinamik faktörler nedeniyle, hattın yeniden düzenlenmesini dikkate alan dinamik bir görüşten bahsetmektedir. Sculli (1984) daha sonra, kullanıcıyla etkileşimli ve MHD problemlerine iyi çözümler getirebilen bir bilgisayar programı geliştirmiştir.

Shin (1990), beklenen toplam maliyeti minimum yapan uygun bir çevrim zamanının olması gerçeğine bağlı olarak, TMS problemler için etkin bir sezgisel geliştirmişlerdir. Sezgisel böyle bir çevrim zamanı için en uygun görev atamalarına imkan vermektedir.

Sarin ve Erel (1990), TMS montaj hattı dengeleme problemleri için, amacı, belirli bir çevrim zamanında tamamlanamayan görevlerden kaynaklanan beklenen tamamlanmama maliyetinin enküçüklenmesi olan bir maliyet modeli geliştirmişlerdir. Bu maliyet fonksiyonu dinamik

programlamayla yürütülmüştür. Sarin ve Erel (1990), büyük ölçekli problemler için de, problemi alt problemlerine ayıran yaklaşık bir prosedür geliştirmişlerdir. Her bir alt problem için, dinamik programlama prosedürü kullanılarak yaklaşık bir sonuç elde edilmektedir. Bu yaklaşık sonuçlar, prosedürün dal ve sınır tipiyle geliştirilmekte ve geliştirilmiş alt problem sonuçları, daha sonra orjinal problemin sonucunu üretmek üzere birleştirilmektedir.

Yukarıda anlatılan metotlara ek olarak Buxey (1978), Driscoll ve Abdel-Shafi (1985), Moberly ve Wyman (1973), Arcus (1966), Silverman ve Carter'ın (1986) TMS dengeleme problemleri üzerine Simülasyon çalışmalarına rastlanmaktadır.

2.3. Çok/Karışık Modelli Deterministik Hat Dengeleme Problemi (ÇMD)

Hem ÇMD ve hem de ÇMS hat dengeleme problemleri için geliştirilen ilk öncü model Arcus (1966) un COMSOAL metodudur. Karışık modeller için model, standart görev zamanlarını farklı modellerin nispi sıklıklarıyla çarpmakta ve bu ortalama zamanları, ÇMD hatların dengelenmesinde bir "Net çevrim zamanı" süresince kullanmaktadır.

Karışık modelli montaj hatları için, modellerin sıralanması problemini ilk olarak Kilbridge ve Wester

(1961) tanımlamışlardır. Onlar, sabit ve değişir oran olmak üzere iki model yükleme sistemi sunmuşlardır.

Thomopoulos 'un (1967, 1970), ÇMD problemler kategorisinde oldukça anlamlı çalışmaları vardır. İlk makalesinde, TMD metodlarını ÇMD hat dengeleme problemlerine uyarlamıştır. Hatların dengelenmesinde, Kilbridge ve Wester'in (1961) sezgisel metodunu geliştirerek, görevlerin çevrim zamanına göre değil de, bir çalışma devresi (shift) esas alınarak operatörlere atanmasını sağlamıştır. Modellerin sıralanmasında ise prosedür, sıralama neticesindeki etkinsizliğin ceza maliyetleriyle ilgilidir. Thomopoulos (1970), görevleri seri tarzdaki istasyonlara atayarak, istasyon atamalarını düzgünleştirmeye çalışmıştır. Metot, her bir istasyon için kabul edilebilir bir kombinasyona ulaşılan kadar, görevlerin uygun kombinasyonlarının araştırılması esasına dayanır.

Roberts ve Villa (1970), her bir iş istasyonunun iş yükündeki fazlalığı enküçükleyen bir tamsayılı programlama formülasyonu sunmuşlar ve modellerin serim (network) versiyonunu optimal bir şekilde çözmek için Gutjahr ve Nemhauser'in (1964) En kısa yol algoritmasını kullanmışlardır.

Macaskill (1972), büyük ölçekli karışık modelli hat

dengeleme problemlerinde, görev gruplarını birleştirerek, bilgisayar yardımıyla atamaları yapan bir sezgisel metot geliştirmiştir. Yine Macaskill (1973), CMD montaj hatları için bir bilgisayar simülasyon çalışması yapmıştır. Simülasyon ya modelin emredilen sırasını kabul etmekte, ya verilen sırayı geliştirmek için otomatik ayarlamalar yapmakta, ya da üretim ihtiyaçlarını karşılamak üzere kendi sırasını üretmektedir.

Dar-El (1978), Dar-El ve Cothar (1975), toplam hat uzunluğunun enküçüklenmesi amacı altında, karışık modellenli hatların dengelenmesi ve modellerin sıralanması için algoritmalar geliştirmişlerdir.

Chakravarty ve Shtub (1985), karışık modellenli hatlar için kurma, stok ve iş gücü maliyetlerinden oluşan toplam üretim maliyetini enküçükleyen bir hat dengeleme yaklaşımı önermişlerdir. Chakravarty ve Shtub'un (1992), karışık modellenli montaj hatlarının çalışmasında öğrenmenin etkisiyle ilgili çalışmaları da vardır.

Johnson (1983) CMD problemler için, iş istasyonların paralelliği ve gerekli görev atamaları gibi bazı pratik faktörleri doğrudan modele ilave etmiştir.

Sumichrast, Russel ve Taylor (1992), bir JIT üretim sistemindeki karışık modellenli montaj hatları için, sıralama

prosedürlerinin karşılaştırmalı bir analizini yapmışlardır. Analizde yer alan sıralama prosedürlerinin performansları da yapılan bir simülasyon çalışmasıyla değerlendirilmiştir.

2.4. Çok/Karışık Modelli Stokastik Hat Dengeleme Problemi (ÇMS)

Arcus (1966) tarafından geliştirilen ve içerisine bir simülasyon modelinin de ilave edildiği COMSOAL' ın değişik bir versiyonu, bu tür problemlere çözüm getirebilmektedir.

Vrat ve Virani (1976), Kottas ve Lau'nun (1973) metodunu gerçek problemlere uygulamışlardır. Vrat ve Virani (1976), Çevrim zamanından büyük görev zamanları için paralel istasyonların kurulmasına imkan verecek şekilde, Kottas ve Lau'nun (1973) metodunu geliştirmişlerdir.

Johnson (1983), bazı pratiğe yönelik kısıtlar altında, istasyon sayısını enküçükleyen bir dal ve sınır algoritması sunmuştur. Bu kısıtlardan bazıları; Stokastik görev zamanları, belirli görevlerin belirli istasyonlara atanması, paralel istasyonlar, gerekli görev bölmeleri vs.

Bu bölümde, MHD problemlerinin değişik sınıfları için kaynak araştırması yapılmıştır. Çalışmada ÇMD MHD problemleriyle ilgilenildiğinden, Bölüm 3 de, bu problemlerle ilgili optimal çözüm metotları detaylı olarak incelenecek ve alternatif modeller geliştirilecektir.

BÖLÜM 3

DETERMİNİSTİK KARIŞIK MODELLİ MHD PROBLEMLERİ İÇİN OPTİMAL ALGORİTMALAR

MHD literatüründeki çalışmaların büyük bir kısmı, tek modellenli MHD problemlerinin dengelenmesiyle ilgilidir. Halbuki birden fazla modelin aynı anda üretim hattında üretilmesi (Bu tür üretime endüstride sıkça rastlanmaktadır.) sırasında ortaya çıkan dengeleme problemleri ile ilgili geliştirilen algoritmaların sayısı oldukça azdır. Bu az sayıdaki algoritmaların çoğu da sezgisel algoritmalarlardır. Problemin yapısındaki karmaşıklık ve birden fazla ürünün aynı anda üretilmesi neticesinde ortaya çıkan zorluklardan dolayı, bu konuda optimal çözüm üreten algoritmaların sayısı yok denecek kadar azdır.

Roberts ve Villa (1970) bu tür problemler için bir tamsayılı programlama modeli geliştirmiş ve yine aynı tür problemlerin çözümü için de En Kısa Yol algoritmasını kullanmışlardır.

Bu bölümde, Roberts ve Villa'nın (1970) bu çalışmaları üzerinde yoğunlaşılacak ve alternatif modeller üretilerek mukayese edilecektir.

3.1. Roberts ve Villa'nın (1970) Tamsayılı Programlama Modeli

Roberts ve Villa (1970), ÇMD montaj hattı dengeleme problemini şöyle tanımlamışlardır:

N ürün ve her birine ait öncelik ilişki diyagramı (kısıtları) verildiğinde;

- i) Her bir görev sadece bir iş istasyonuna atanacak,
- ii) İstasyon sayısı bütün ürünler için aynı olacak,
- iii) Öncelik ilişkileri sağlanacak,
- iv) Herhangi bir ürün için herhangi bir istasyonun iş yükü çevrim zamanını geçmeyecek,

$$S_{ki} \leq C \quad \begin{matrix} i=1, \dots, N \\ k=1, \dots, K \end{matrix}$$

S_{ki} : i ürünü için k istasyonunun iş yükü
ve

- v) Toplam boş zaman enküçüklenecek,
- şekilde görevlerin iş istasyonlarına atanmasıdır.

Modelde kullanılan notasyonlar:

$$X_{ki}^j = \begin{cases} 1 & \text{i ürününe ait j görevi k istasyona atanırsa} \\ 0 & \text{Diğer durumlarda} \end{cases}$$

$$t_i^j = \text{i ürününe ait j görevinin zamanı}$$

$$J_i = \text{i ürünündeki görev sayısı (seti)}$$

i ürünü için minimum istasyon sayısı K_i^* olarak ifade edilmiştir.

$$K_i^* = \left\lceil \sum_{j=1}^{J_i} t_i^j / C \right\rceil$$

[x]: x'e eşit ya da ondan büyük, enküçük tamsayı.

Modelde, ürünlerin C değeri ve iş yükü için

$$K_i^* = K^* \quad i=1, \dots, N$$

varsayımını yapmışlardır.

Bu nedenle, hattı K^* dan küçük bir istasyon sayısı ile dengelemek mümkün olmadığından, istasyon sayısını enküçükleme için K^* dan sonraki iş yükünü enküçükleme yeterli olacaktır. Eğer bir istasyonun (k_1) iş yükü ($k_1 > K^*$) lar için sıfır ise, o zaman bütün istasyonların ($k > k_1$) iş yükü sıfır olmak zorundadır.

Amaç Fonksiyonu;

$$\text{Min } f = \sum_{k=K^*+1}^M W_k \sum_{i=1}^N n_i \sum_{j=1}^{J_i} X_{ki}^j$$

f : Toplam boş zaman

n_i : i ürünün birim sayısı

W_k ' nın amacı, $k_1 < k_2$ olduğunda, k_1 istasyonuna, k_2 ye nazaran daha fazla öncelik vererek atamaların yapılmasını

sağlamaktır. W_k değerleri artan pozitif katsayılardır.

$$W_k = 2^{k-(K^*+1)} \quad k = K^*+1, \dots, M$$

M sayısı, M den fazla iş istasyonlarına, görevlerin atanmamasını temin edecek şekilde belirlenen, yeteri kadar büyük bir sayıdır.

M için kabul edilebilir bir seçim;

$$M = \max_i \{J_i\}$$

olarak belirlenmiştir.

Roberts ve Villa'nın (1970) tamsayılı programlama modelindeki kısıtlar aşağıda verilmiştir:

a) Her bir görev sadece bir kez atanmalıdır.

$$\sum_{k=1}^M X_{ki}^j = 1 \quad \begin{matrix} j=1, \dots, J_i \\ i=1, \dots, N \end{matrix}$$

b) İstasyon sayısı, bütün ürünler için aynı olmalıdır. Yani; eğer i_1 ürünü için k istasyonunun iş yükü sıfır ise, o zaman diğer bütün ürünler için bu istasyonun iş yükü de sıfır olmak zorundadır.

$$\sum_{j=1}^{J_{i_1}} X_{ki_1}^j \leq M \sum_{j=1}^{J_{i_2}} X_{ki_2}^j \quad \begin{matrix} i_1=1 \\ i_2=2, \dots, N \\ k=1, \dots, M \end{matrix}$$

c) Öncelik ilişkileri sağlanmalıdır. Eğer i ürününün j_1 ve j_2 görev için; j_1 , j_2 den önce geliyorsa $j_1 \prec j_2$ yazılabilir.

$$\sum_{k=1}^M k \cdot X_{ki}^{j_1} \leq \sum_{k=1}^M k \cdot X_{ki}^{j_2} \quad \begin{matrix} i=1, \dots, N \\ \text{tamamı için } j_1 \prec j_2 \end{matrix}$$

d) İstasyonların iş yükü, çevrim zamanını geçemez.

$$\sum_{j=1}^{J_i} t_{ki}^j \leq C \quad \begin{matrix} k=1, \dots, M \\ i=1, \dots, N \end{matrix}$$

$$e) \quad X_{ki}^j \in \{0, 1\} \quad \begin{matrix} j=1, \dots, J_i \\ i=1, \dots, N \\ k=1, \dots, M \end{matrix}$$

$K_i^* = K^*$ varsayımı sağlanmadığında ise

$$K^* = \max_i \{K_i^*\}$$

şeklinde seçilebilir.

Bölgeleme kısıtları da ayrıca ifade edilebilir. Örneğin, i_1 ürününün j_1 ve i_2 ürününün j_2 görevi teknolojik kısıtlamalardan dolayı aynı istasyonda yapılmak zorunda ise, o zaman kısıt,

$$X_{ki}^{j_1} = X_{ki}^{j_2} \quad k=1, \dots, K_1$$

şeklinde ifade edilebilir.

Bu formülasyon, orta büyüklükteki bir problem için dahi, büyük değişken ve kısıt sayısına sebep olmaktadır. Bu nedenle bu formülasyonun pratik ilgiden ziyade teorik olduğu ifade edilmiştir. Roberts ve Villa'nın (1970) bu formülasyonu geliştirmelerindeki amaçlardan birisi, 0-1 tamsayılı programlamada hesaplama etkinliğinin artırılması yönünde çalışmaların yapılması ve ileride bu yapıdaki bir modelin faydalı olabileceğinin düşünülmesidir.

Not: Bu kısımda kullanılan notasyonların tamamı, orjinal çalışmadakilerle aynıdır.

3.2. Geliştirilen Yeni 0-1 Tamsayılı Programlama Modeli

Burada, karışık modellenli deterministik MHD problemleri için Roberts ve Villa'nın (1970) formülasyonuna göre daha etkili yeni bir 0-1 tamsayılı programlama modeli geliştirilmiştir.

Montaj üretimi yapan sanayiilerin çoğunda, aynı ürünün benzer tiplerinin üretiminin yapıldığını görmekteyiz (örneğin; Otomobil sanayi). Dolayısıyla bu tür üretim hatlarında ürün tipleri benzer olduğu için, üretim hattında

yapılan işlemlerin çoğunun ortak olduğunu da söylemek mümkündür. Geliştirilen bu modelin en önemli varsayımı da benzer tiplerin görevlerinin büyük bir kısmının ortak olmasıdır. Ayrıca, benzer tiplerin bu ortak görevlerinin aynı istasyona atanması gerektiği de işletme açısından büyük faydalar sağlayacaktır.

Geliştirilen modelin varsayımları aşağıda verilmiştir:

- 1- Birden fazla ürün, aynı üretim hattında üretilmektedir.
- 2- Her bir ürünle ilgili görev zamanları deterministiktir.
- 3- Her bir ürünün öncelik ilişkileri bilinmektedir.
- 4- Aynı üretim hattında üretilen benzer modellerin görevlerinin çoğu ortaktır.
- 5- İstasyonlar arası ara stoğa izin verilmemektedir.
- 6- Benzer modellerin (tiplerin) ortak görevleri aynı istasyonlara atanmalıdır.
- 7- Tüm modeller için istasyon sayıları aynıdır.
- 8- Paralel istasyonlara mücade edilmemektedir.

Modelimizde, ürünlerin ortak görevlerinin olması durumu dikkate alındığından, her bir ürüne ait öncelik diyagramlarını, tek bir diyagram (Combined diyagram) şeklinde ifade etmek mümkündür. Bu da problemdeki kısıt ve değişken sayısını büyük ölçüde azaltacaktır.

Problem formülasyonuna geçilmeden önce, her bir modele ait öncelik diyagramlarının (1.6.4) de ifade edildiği gibi, ortak bir öncelik diyagramında birleştirilmeleri gerekmektedir.

Patterson ve Albracht'ın (1975), tek modellenli hatların tamsayılı programlama formülasyonunda kullandıkları E_i (En erken istasyon) ve L_i (En geç istasyon) hesaplamaları, problemdeki değişken sayılarını sınırlandırmıştır.

Patterson ve Albracht'ın (1975) bu istasyon sınır hesaplamaları, gerekli şekilde düzenlenerek, karışık modellenli montaj hatlarının dengelenmesi için geliştirilen 0-1 tamsayılı formülasyonda kullanılmıştır. E_i ve L_i üzerinde yapılan düzenleme, üretim hattında üretilen ürünlerin benzer olması (ortak elemanlarının çok olması) durumunda anlamlı ve geçerlidir. Aksi halde, yani ortak görevleri olan ancak fazla olmayan durumlarda ise, her bir ürün için ayrı ayrı E_i ve L_i değerlerinin bulunması ve bu değerlerin birleşimi alınmak kaydıyla elde edilen değerlerin kullanılması yoluna gidilebilir.

Farklı modellerin birleştirildiği ortak öncelik diyagramı üzerinde, her bir görev için E_i ve L_i değerlerinin hesaplanması gerekmektedir. E_i ve L_i değerlerinin hesaplanmasıyla ilgili ifadeler aşağıda verilmektedir

$$E_i = \begin{cases} 1 & \text{Eger } (\sum_{m \in MS_i} t_{im} + \sum_{\substack{j \in P_i \\ m \in MS_j}} t_{jm}) = 0 \quad i=1, \dots, \Lambda \\ [(\sum_{m \in MS_i} t_{im} + \sum_{\substack{j \in P_i \\ m \in MS_j}} t_{jm}) / \sum_{m=1}^P c_m]^+ & \text{diğer durumlarda} \end{cases}$$

$$L_i = \begin{cases} K & \text{Eger } (\sum_{m \in MS_i} t_{im} + \sum_{\substack{j \in S_i \\ m \in MS_j}} t_{jm}) = 0 \quad \text{ise} \\ K+1 - [(\sum_{m \in MS_i} t_{im} + \sum_{\substack{j \in S_i \\ m \in MS_j}} t_{jm}) / \sum_{m=1}^P c_m]^+ & \text{d.d} \end{cases}$$

$[x]^+$: x'e eşit yada x ten büyük enküçük tamsayı.

Bu şekilde E_i ve L_i değerlerinin hesaplanması, her bir görevin sadece bu değerler arasındaki istasyonlara atanmasını temin edecektir. 0-1 tamsayılı programlama modelinin ifade edilmesi ve program WKM 'ye girdi olarak verilmesi bakımından bu değerlerin kolayca hesaplanmasını sağlayan yazılım, örnek bir problem için çözüm çıktısıyla birlikte EK-1 de verilmiştir.

E_i ve L_i değerlerinin hesaplanmasında mümkün en fazla istasyon sayısı K' nın belirlenmesi gereklidir. K' nın belirlenmesinde bir yol, en fazla görev sayısına sahip modelin görev sayısının dikkate alınmasıdır. Ancak

hesaplama avantajı bakımından K değerinin mümkün olduğu kadar küçük belirlenmesinde fayda vardır. Çünkü, 0-1 değişkenlerin ve kullanılan kısıtların sayısı, büyük ölçüde öncelik ilişkilerin yapısına, K' ya ve N 'ye bağlıdır. Bu nedenle ileriki bölümlerde ÇMD hatlar için geliştirilecek olan bir sezgisel metodun belirlediği istasyon sayısını (K) bu formülasyonda ve modelin diğer bütün hesaplamalarında kullanmak mümkündür ve büyük avantaj sağlayacaktır.

Bütün bu açıklamalardan sonra, geliştirilen 0-1 modelin kısıtları sırasıyla aşağıda ifade edilmiştir:

Atama Kısıtları

Atama kısıtları, her bir modeldeki görevlerin en fazla bir istasyona atanmasını temin eder. Bu kısıtlar aşağıdaki gibi yazılabilir:

$$\sum_{k \in E_i}^{L_i} V_{ik} = 1 \quad i=1, \dots, N$$

Eğer $V_{ik}=1$ ise, Birleştirilmiş diyagramda i görevine (düğüme) sahip bütün modeller için, i görevi/düğüme k istasyonuna atanır.

Öncelik Kısıtları

Bu kısıtlar, birleştirilmiş öncelik diyagramına göre oluşturulurlar.

$$\sum_{k=E_a}^{L_a} k \cdot V_{ak} - \sum_{k=E_b}^{L_b} k \cdot V_{bk} \leq 0$$

Birleştirilmiş diyagramdaki her bir ilişki için $a \ll b$ ve $L_a \geq L_b$ dir.

Problemde model sayısı kaç tane olursa olsun, modelimizdeki atama ve öncelik kısıtları sayısı, sırasıyla birleştirilmiş diyagramdaki düğüm sayısı ve diyagramdaki dal sayısı kadar olacaktır.

Çevrim Zamanı Kısıtları

Herhangi bir modelin görevlerinin atandığı bir istasyonun iş yükü (o istasyondaki görev zamanlarının toplamı), o modele ait çevrim zamanını geçmemelidir.

$$\sum_{i \in W_{km}} t_{im} \cdot V_{ik} \leq C_m \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, P \end{matrix}$$

İş İstasyon Kısıtları

Montaj hattında üretilen modellerin tümü için, istasyon sayısı aynı olmalıdır. Eğer birinci ürün için k istasyonunun iş yükü sıfır ise, diğer bütün ürünler için de

bu istasyonun iş yükü sıfır olmalıdır. Bu durum, aşağıdaki kısıtlarla sağlanabilir:

$$\sum_{i \in W_{km}} V_{ik} - |W_{km}| X_{km} \leq 0 \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, P \end{matrix}$$

$$\sum_{m=1}^P X_{km} - P \cdot A_k = 0 \quad k=1, \dots, K$$

X_{km} bir 0-1 değişkendir. Eğer her hangi bir m modeline ait bir görev, k istasyonuna atanmışsa (k istasyonu m ürünü için kullanılıyorsa), X_{km} 1'e eşittir, diğer durumlarda ise sıfırdır. A_k değişkeni de bir 0-1 değişkendir. Eğer k istasyonu bütün ürünler tarafından kullanılıyorsa A_k 1'e, diğer durumlarda 0'a eşittir. Başka bir ifadeyle, eğer bütün X_{km} ler k istasyonu için 1'e eşitse; A_k da 1'e eşittir.

Amaç Fonksiyonu

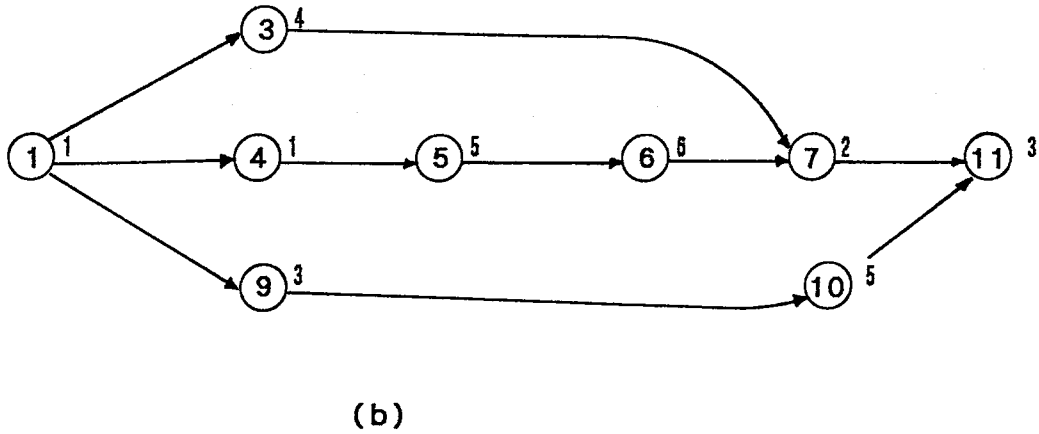
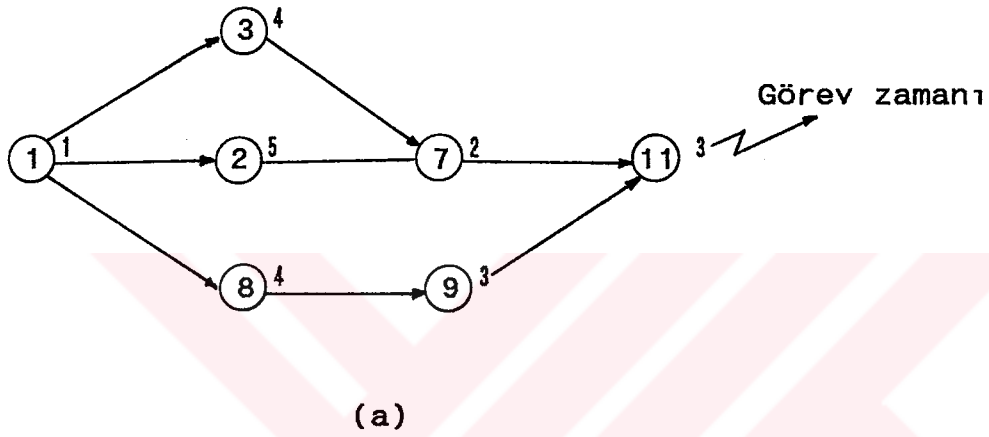
Amaç fonksiyonu, aşağıda gösterildiği gibi, istasyon sayısının enküçüklenmesini hedeflemektedir.

$$\text{Min} \sum_{k=1}^K A_k$$

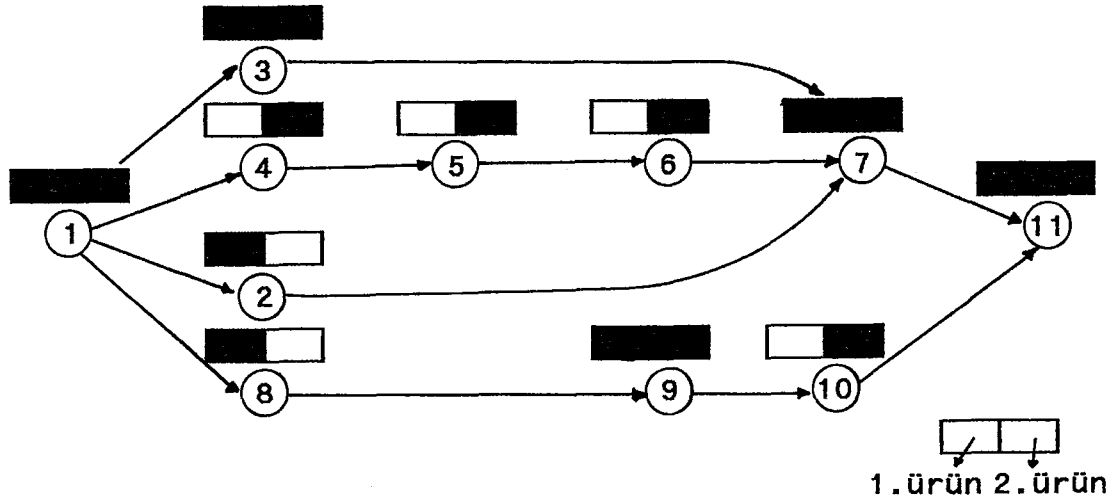
3.2.1. Örnek Problem

Bu kısımda, karışık modellenli MHD problemleri için

geliştirilen 0-1 tamsayılı programlama formülasyonu bir örnek probleme uygulanacaktır. Problemde iki modelin aynı üretim hattında üretilmesi söz konusudur. Her bir modele ait öncelik diyagramları ve birleştirilmiş diyagram sırasıyla Şekil 3.1 ve Şekil 3.2 de verilmiştir.



Şekil 3.1 İki Ürünlü Örnek Problem için Öncelik Diyagramları. (a) 1. Ürün (b) 2. Ürün



Şekil 3.2 İki Ürünlü Örnek Problem için Birleştirilmiş Öncelik Diyagramı.

Hangi düğümde, hangi modellerin olduğunu, başka bir ifadeyle herhangi bir düğümün hangi modeller tarafından paylaşıldığını kolayca görebilmek için diyagramda düğümler üzerine bölmeli kutular koyulmuştur. Bu kutuların her bir bölmesi, bir modeli göstermektedir. Bölmelerin taranmış olması, o bölmede bulunan modelin, ilgili düğüme sahip olduğunu göstermektedir.

Problemde, her bir modele ait çevrim zamanı $C=10$ dk ve mücade edilen maximum istasyon sayısı da $K=4$ olarak belirlenmiştir. Amaç, en az sayıda istasyon ile üretim hattının dengelenmesidir.

Problem LINDO [1988] paket programı kullanılarak

çözölmüş ve optimal atamalar Tablo 3.1 de verilmiştir.

Problemin çözümü neticesinde elde edilen V_{ik} değeri şöyledir: $V_{11}=1$, $V_{41}=1$, $V_{51}=1$, $V_{81}=1$, $V_{91}=1$, $V_{32}=1$, $V_{62}=1$, $V_{23}=1$, $V_{73}=1$, $V_{103}=1$, $V_{113}=1$, diğeri V_{ik} lar =0 .

1. istasyon için 1 ve 9 nolu görevler, 2. istasyon için 3 nolu görev ve 3. istasyon için 7 ve 11 nolu görevler ortak görevlerdir (Bu durum Şekil 3.2' deki birleştirilmiş öncelik diyagramındaki kutulardan da görölebilir). Yani hem 1. hem de 2. model, bu görevlere sahiptirler. Dolayısıyla bu ortak görevler hem 1. ürün hem de 2. ürün için ilgili istasyonlara atanırlar.

Tablo 3.1 Örnek Problem için Optimal İstasyon Atamaları

İstasyon	Görevler	1. Ürün		2. Ürün	
		Görevler	İstasyon zamanı	Görevler	İstasyon zamanı
1	1,4,5,8,9	1,8,9	8	1,4,5,9	10
2	3,6	3	4	3,6	10
3	2,7,10,11	2,7,11	10	7,10,11	10

Örnek problemle ilgili detaylar (formölasyon kısıtları) EK-2 'de verilmiştir. Modelimize göre gerekli kısıt sayısı 39, gerekli değişken sayısı da 38 olarak belirlenmiştir.

3.3. Modellerin Karşılaştırılması

Geliştirilen 0-1 modeli, farklı boyuttaki problemler için denenmiş ve gerekli kısıt ve değişken sayıları Tablo 3.2 'de verilmiştir. Denenen problemlerin hepsi, ortak iş elemanlarına sahip problemlerdir ve çevrim zamanları her bir problemdeki bütün ürünler için aynı alınmıştır.

Tablo 3.2 Farklı Büyüklükteki Problemler için iki Modelin Karşılaştırması

Ürün Sayısı	Toplam Görev Sayısı	Toplam Öncelik İlişki Sayısı	Birleş. Diyag.ki Görev Sayısı	Birleş. Diyag.daki İlişki Sayısı	Model I Geliştirilen		Model II R&V 'nın	
					Gerekli Kısıt Sayısı	Gerekli Değişken Sayısı	GKS	GDS
2	9	8	5	6	26	19	32	45
2	16	18	11	13	39	38	61	144
3	51	57	30	36	122	157	208	1020
3	61	72	28	34	125	173	248	1403
4	39	46	16	21	91	103	169	468
4	56	67	23	31	126	153	242	952

Tablodan görüleceği gibi geliştirilen 0-1 modelinin sahip olduğu kısıt ve değişken sayıları, Roberts ve Villa'nın (1970) modeline kıyasla oldukça azdır. Bunun sebebi ise, geliştirilen 0-1 modelde Birleştirilmiş öncelik diyagramının kullanılması, maksimum istasyon sayısının mümkün olduğu kadar küçük sayıda tutulması (sezgisel algoritmadan temin edilmesi) ve E_i ve L_i hesaplamalarının

düzenlenerek modele ilave edilmesidir.

Geliştirilen yeni 0-1 tamsayılı programlama modeliyle, orta ölçekli (30-60 görev) ÇMD montaj hattı dengeleme problemlerinin kısıt ve değişken sayılarını belirli bir sayıda tutmak mümkündür. Ancak MHD problemleri NP karakteristiğe sahip olduğundan, kısıt sayısı arttıkça çözüm zamanı da üstel olarak artacaktır. Dolayısıyla her ne kadar modelimiz pratiğe uygun varsayımlar taşısa da, büyük boyutlu problemler için, modelin çözümü zorlaşacaktır. Bu nedenle kısıt ve değişken sayısını daha da küçültmek ve dolayısıyla bilgisayar zamanını kabul edilebilir bir seviyede tutmak için problemin ayrıştırılması ya da kısıt bakımından rahatlatılması gereklidir.

3.4. En Kısa Yol Algoritması

Robert ve Villa (1970), çok ürünli MHD problemleri için 0-1 tamsayılı programlama modelinin yanısıra, bir de Serim (Network) Modeli sunmuşlardır. Bu kısımda Robert ve Villa (1970) tarafından geliştirilen serim modeli işlenecek ve prosedür detaylı olarak açıklanacaktır. Model ifade edilirken kullanılacak olan notasyonlarda, tamamen orjinaline sadık kalınmıştır.

Robert ve Villa (1970) modellerini geliştirirken, Gutjahr ve Nemhauser 'in (1964) tek modelli MHD problemleri

için geliştirdikleri En kısa yol (shortest route) algoritmasından faydalanmışlardır.

Bir serimde, düğümler durumları (istasyon atamalarını) oklar (arcs) ise istasyonları göstermektedir. Ok uzunlukları, istasyonlardaki boş zamanı ifade etmektedir. Böylece optimizasyon prosedürü, serimde bir en kısa yolun bulunması şeklindedir.

Çok modellenli MHD problemleriyle ilgili serim'in geliştirilmesinde kullanılan düğümler (durumlar) ve okların üretilmesi aşağıda gösterilmiştir.

3.4.1. Durumlar ya da düğümlerin üretilmesi

Bir Durum (state), uygun ilk istasyon atamalarını teşkil eden görevlerin sıralı bir setidir. Modelde, ϕ durum hariç, her bir ürün için tüm durumların üretilmesi gereklidir. Her bir i ürünü için, S_i durum seti oluşturulur. Çok ürünlü problemler için durumların seti, tüm S_i lerin Kartezyen çarpımından elde edilebilir.

$$S = \{S_1 \otimes S_2 \otimes S_3 \otimes \dots \otimes S_N, \phi\}$$

S deki sembolik elemanlara T_i diyelim.

$$T_1 = \{S_{11}, S_{21}, \dots, S_{M1}\}$$

S_{ij} , S_i ($i=1, \dots, N$) nin bir elemanıdır. Eğer bölgesel sınırlamalar söz konusuysa, o zaman bu şartı bozan durumlar hemen çıkarılabilirler. Düğümler, ϕ durumdan (düğümden)

oluşturulmaya başlanır. Serimdeki son durum (düğüm) r , tüm ürünlerin elemanlarından oluşmaktadır. Modeldeki durum üretimi, Gutjahr ve Nemhauser 'in (1964) modelinden oldukça farklıdır.

3.4.2. Yönlü okların yapılanması

Eğer $S_{iu} \subset S_{iv}$ $i=1, \dots, N$ ve

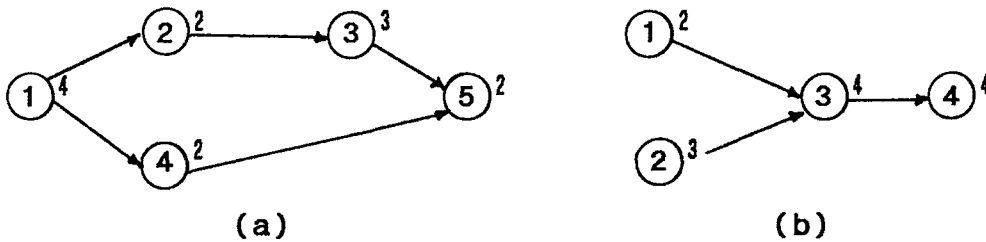
$$\sum_{j \in S_{iv} - S_{iu}} t_j^j \leq C \quad i=1, \dots, N \text{ ise}$$

T_u düğümünden, T_v ye bir yönlü ok var demektir.

Serim bu şekilde oluşturulduktan sonra, o dan r ye en kısa yolun bulunması optimal istasyon atamalarının bulunması bakımından yeterlidir. Modelin işleyişi aşağıdaki örnek problem üzerinde açıklanmıştır.

3.4.3. Örnek problem

İki ürüne ait öncelik diyagramları Şekil 3.3 'deki gibi verilmiş olsun ve her bir ürün için çevrim zamanı da 5 olarak belirlensin.



Şekil 3.3 Öncelik Diyagramları (a) 1.ürün (b) 2.ürün

Öncelik diyagramlarından, her bir ürüne ait durumlar oluşturulur.

$$S_1=\{(1),(1,2),(1,4),(1,2,3),(1,2,4),(1,2,3,4),(1,2,3,4,5)\}$$

$$S_2=\{(1),(2),(1,2),(1,2,3),(1,2,3,4)\}$$

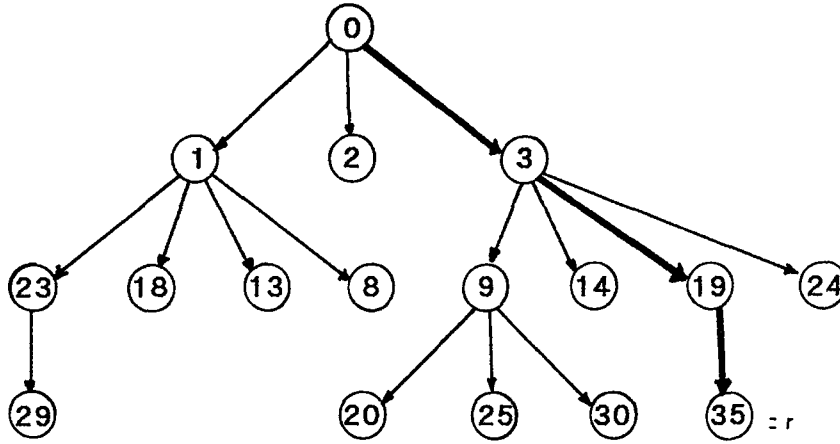
İki durum setinin kartezyen çarpımlarıyla, durumların seti elde edilir.

$$S=\{[(1),(1)], [(1),(2)], [(1),(1,2)], [(1),(1,2,3)], [(1), (1,2,3,4)], \dots, [(1,2,3),(1)], \dots, [(1,2,3,4,5),(1,2,3,4)], \emptyset\}$$

Oluşturulan durumların detaylı listesi Tablo 3.3 'de, Durumlara göre elde edilen serim ise, Şekil 3.4 'de verilmiştir.

Tablo 3.3 Durum Üretimi

Düğüm No.	Durum Elemanları		Durum Zamanı	
	1.Ürün	2.Ürün	1.Ürün	2.Ürün
0			0	0
1	1	1	4	2
2	1	2	4	3
3	1	1,2	4	5
4	1	1,2,3	4	9
5	1	1,2,3,4	4	13
6	1,2	1	6	2
7	1,2	2	6	3
8	1,2	1,2	6	5
9	1,2	1,2,3	6	9
10	1,2	1,2,3,4	6	13
11	1,4	1	6	2
12	1,4	2	6	3
13	1,4	1,2	6	5
14	1,4	1,2,3	6	9
15	1,4	1,2,3,4	6	13
16	1,2,3	1	9	2
17	1,2,3	2	9	3
18	1,2,3	1,2	9	5
19	1,2,3	1,2,3	9	9
20	1,2,3	1,2,3,4	9	13
21	1,2,4	1	8	2
22	1,2,4	2	8	3
23	1,2,4	1,2	8	5
24	1,2,4	1,2,3	8	9
25	1,2,4	1,2,3,4	8	13
26	1,2,3,4	1	11	2
27	1,2,3,4	2	11	3
28	1,2,3,4	1,2	11	5
29	1,2,3,4	1,2,3	11	9
30	1,2,3,4	1,2,3,4	11	13
31	1,2,3,4,5	1	13	2
32	1,2,3,4,5	2	13	3
33	1,2,3,4,5	1,2	13	5
34	1,2,3,4,5	1,2,3	13	9
35=r	1,2,3,4,5	1,2,3,4	13	13



Şekil 3.4 Örnek Problem için Nihai Serim

Şekil 3.4 'deki serimde, o dan r ye en kısa yol bulunarak, optimal istasyon atamaları elde edilir. Her bir istasyondaki görevler aşağıdaki işlemlerle sağlanabilir: 35. Durum, en son durumdur. 35. duruma 19. durumdan, 19.duruma 3. durumdan, 3. duruma ise o. durumdan erişilmektedir. Buna göre;

(35-19) $\{1,2,3,4,5\}-\{1,2,3\}=\{4,5\}$: 1.ürün için

$\{1,2,3,4\}-\{1,2,3\}=\{4\}$: 2. ürün için

(19-3) $\{1,2,3\}-\{1\}=\{2,3\}$: 1. ürün için

$\{1,2,3\}-\{1,2\}=\{3\}$: 2. ürün için

(3-0) $\{1\}-\{\phi\}=\{1\}$: 1. ürün için

$\{1,2\}-\{\phi\}=\{1,2\}$: 2. ürün için

görev setleri oluşturulabilir. Elde edilen optimal istasyon atamaları Tablo 3.4 'de verilmiştir.

Tablo 3.4 Optimal İstasyon Atamaları

İstasyon Sayısı	Görevler	
	1. ürün	2. ürün
1	1	1,2
2	2,3	3
3	4,5	4

3.5. Geliştirilen Serim Modeli

Bu kısımda, karışık modelli montaj hatlarının dengelenmesi problemleri için bir serim modeli sunulmuştur. Model, Robert ve Villa 'nın (1970) geliştirdiği serim modelinde de olduğu gibi, tamamen Gutjahr ve Nemhauser 'in (1964) en kısa yol algoritmasına dayanmaktadır. Ancak Roberts ve Villa 'nın (1970) modeli, aynı üretim hattında üretilen ürünlerin ortak görevlere sahip olabileceği ve bu görevlerin aynı istasyonlara atanmaları gerektiği durumunu dikkate almadığından, modeldeki durumların (state) seti, her bir ürüne ait durumların kartezyen çarpımıyla elde edilmektedir ki, bu durum, 2 den fazla ürünün olduğu küçük boyutlu bir problemde dahi çözümü hemen hemen imkansız hale getirmektedir. Ayrıca elde edilen çözüm, gerçek durumu (pratiği) temsil etmesi bakımından da pek yeterli sayılmamaktadır. Bu nedenle geliştirilen serim modelinde, bu gerçekçi faktörler de dikkate alınmış ve model yeniden düzenlenmiştir. Bu modelde, daha önce geliştirilen 0-1 tamsayılı programlama modelinde olduğu gibi ürünlerin çok

benzer olmasına, diğ er bir ifadeyle, ortak g revlerin fazla olması zorunluluėu yoktur. En az 1 ortak g revin olması, modelin i letilmesi i in yeterlidir.

Model, Gutjahr ve Nemhauser 'in (1964) TMD problemler i in geli tirilen en kısa yol algoritmasıyla b y k  l  de aynıdır. Sadece, algoritma  zerinde, birden fazla  r n dikkate alınabilecek  ekilde bazı ayarlamalar yapılmı tır.

 imdi, modelle ilgili gerekli a ıklamalar yapılacak ve model,  rnek bir problem  zerinde  alı tırılacaktır.

Bir Durum (state), istasyon atamaları  eklinde dikkate alınabilen, t m g revlerden olu an R setinin bir alt setidir. DRM_i , $i=0,1,\dots,r$ durumlarını ifade eder. Ba langı  durumu $DRM_0=\phi$ ve en son durum da $DRM_r=R$ ile ifade edilir. DRM_i durumdaki her bir  r ne ait g revlerin zamanlarının toplamı a aėıdaki e itliklerle belirlenebilir:

$$\begin{aligned} t_m(DRM_0) &= \phi & m=1, \dots, P \\ t_m(DRM_i) &= \sum_{x \in DRM_i} t(x) & \begin{matrix} m=1, \dots, P \\ i=1, \dots, r \end{matrix} \end{aligned}$$

Bu hesaplamalar, birle tirilmi   ncelik diyagramına g re yapılmaktadır. Bir SR seriminde durumlar, d ė mlerle ifade edilmektedir. Yani i d ė m ne kar ılık gelen DRM_i durumudur.

Bir SR seriminde, eğer,

$$DRM_i \subset DRM_j \text{ ve } t_m(DRM_i) - t_m(DRM_j) \leq C_m, m=1, \dots, P$$

ise (tüm düğümler için aynı anda), i düğümünden, j düğümüne bir yönlü (directed) (ij) oku var demektir. Serimdeki her yönlü ok (ij),

$$\sum_{m=1}^P [C_m - t_m(DRM_j) - t_m(DRM_i)]$$

uzunluğuyla atanmaktadır. Hiç bir ok, boş sete karşılık gelen o düğümüne giremez ve yine hiç bir ok, tüm görevlerin setine karşılık gelen r düğümünden ayrılamaz. Dolayısıyla SR serimi, üzerinde tüm durumların setini barındıran o dan r ye yönlü sonlu bir serimdir.

Bir SR seriminde, o dan r ye en kısa yol şu şekilde yapılabilir;

1- o düğümünden başlanır ve tüm oklar bu düğümden dallanır. Eğer $t_m(DRM_i) \leq C_m$ $m=1, \dots, P$ ise, bir oi oku mevcut demektir. Erişilen düğümler, ilk düğümler olarak adlandırılırlar.

2- s. düğümlere bir defada ulaşıldığı varsayılınsın. Eğer $DRM_j \subset DRM_k$ ve $t_m(DRM_k) - t_m(DRM_j) \leq C_m$ ise, s. düğümler arasındaki her bir j düğümü için, k düğümüne bir ok gönderilir. s. düğümlerden erişilen düğümlere (s+1). düğümler denilir. s+1 den r düğümüne bir defada erişilene kadar adım 2 tekrar edilir.

3.5.1. Durumların Üretimi

Görev setleri üretilirken, aşağıdaki esasların sağlanması gerekmektedir.

- i) Durum üretiminde hiç bir durum elemanı, tekrarlı olarak üretilemez.
- ii) Üretilen tüm setler, durumlardır.

İlk durum üretimi olarak boş set dikkate alınır. Birleştirilmiş öncelik diyagramında kendisinden önce elemanı olmayan görevler, 1. aşamada (stage) yer alırlar. Eğer bunlara DS eleman seti dersek, DS nin her bir alt seti bir durumdur. 1. aşamanın sonunda, DS nin elemanları işaretlenir, diğerleri ise işaretlenmemiş görevler olarak gözönüne alınır.

Bir DRM durumunun hemen takipçileri (immediate followers), DRM deki elemanların en az 1 tanesinin hemen takipçisi ve DRM de bulunmayan herhangi bir elemandan sonra gelmeyen elemanlar olarak tanımlanırlar. Yani bu elemanlar, öncelik ilişkilerine bakıldığında, direkt olarak istasyon ataması için engeli olmayan ve daha önceki aşamalarda işaretlenmemiş olan elemanlardır.

k aşamasındaki herhangi bir DRM durumu için, DRM nin işaretsiz hemen takipçileri, F(DRM) adı verilen bir listede

yer alırlar ($F(DRM)$: işaretsiz hemen takipçilerin listesi). Her bir alt set için $S \subset F(DRM)$, $DRM \cup S$, $k+1$. aşamada yer alan bir durumdur. k . aşamadaki tüm durumlar gözönüne alındığı zaman, k aşamasındaki tüm DRM ler için $F(DRM)$ deki her bir eleman işaretlenir ve proses $k+1$. aşama için tekrarlanır. Tüm elemanlar işaretlendiğinde, yapılanma tamamlanmış demektir.

İşaretsiz elemanların belirlenmesinde, $k+1$. aşamadaki tüm durumlar için hemen takipçiler ayrı ayrı yazılırken, $k+1$ ve daha önceki aşamalarda işaretlenen görevler dikkate alınmazlar. Sadece, daha önce işaretlenmemiş görevlerin içinden, öncelik ilişkileri dikkate alınarak belirleme yapılabilir.

Durumlar üretilip, serim oluşturulduktan sonra, oklar üzerindeki uzaklıklar dikkate alınarak o dan r ye bir en kısa yolun bulunmasıyla gerekli istasyon atamaları sağlanabilir. o dan r ye bulunan en kısa yol $(o, i, j, k, \dots, r)$ ise,

$$\{DRM(r)-DRM(k), DRM(k)-DRM(j), DRM(j)-DRM(i), DRM(i)-DRM(o)\}$$

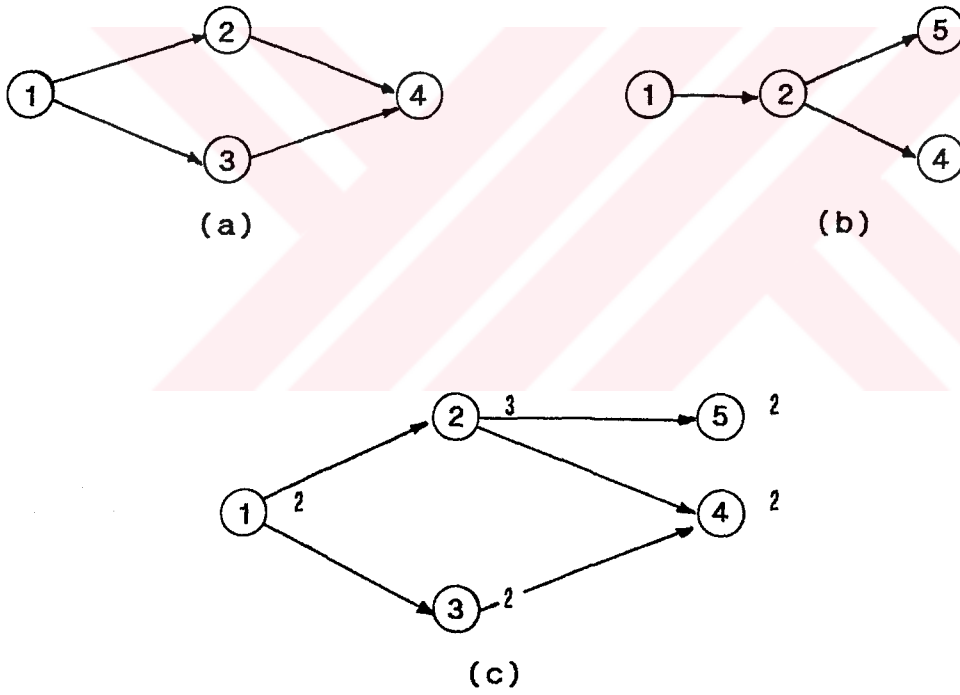
ifadesindeki her bir set grubu, sırasıyla istasyon atamalarını vermektedir.

Bütün bu açıklamalardan sonra, model örnek bir

problem üzerinde çalıştırılmış ve optimal istasyon atamaları elde edilmiştir.

3.5.2. Örnek problem

İki ürüne ait öncelik diyagramları ve birleştirilmiş öncelik diyagramı Şekil 3.5 'de verilmiştir. Her bir ürüne ait çevrim zamanı da 5 olarak belirlenmiştir.



Şekil 3.5 Öncelik Diyagramları (a) 1.ürün
(b) 2.ürün (c) Birleştirilmiş

Tablo 3.5 Durumların Üretilmesi

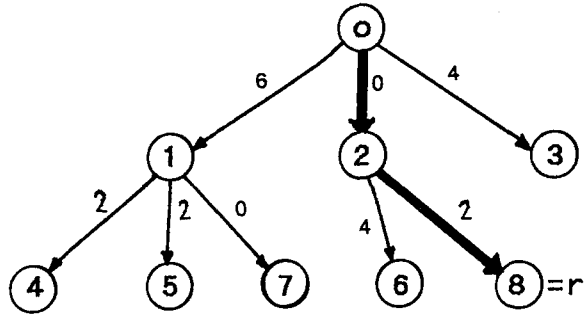
Aşama	Etiket- lenmiş Görevler	Durum No.	Durum Elemanları	Durum zamanı		Etiketsiz hemen takipçiler
				1.ürün	2.ürün	
0		0	Boş set	0	0	1
1	1	1	1	2	2	2,3
2	2,3	2	1,2	5	5	5
		3	1,3	4	2	4,5
		4	1,2,3	7	5	
3	4,5	5	1,2,5	5	7	
		6	1,2,3,4	9	7	
		7	1,2,3,5	7	7	
		8=r	1,2,3,4,5	9	9	

Tablo 3.5 'de her bir durum üretildikten sonra, yapılacak olan en kısa yol hesaplamaları Tablo 3.6 'da verilmiştir.

Tablo 3.6 En Kısa Yol Hesaplamaları

Düğüm No.	Düğümlere Oklar
0	1,2,3
1	4,5,7
2	6,8
3	

Hesaplama sonucunda elde edilen düğüm ve okların serimle gösterilmesi Şekil 3.6 'da verilmiştir.



Şekil 3.6 Örnek Problem için Nihai Serim

1-4 oku üzerindeki uzaklık rakamı şöyle hesaplanmıştır.

$$C_1 = (t_1(\text{DRM}(4)) - t_1(\text{DRM}(1))) +$$

$$C_2 = (t_2(\text{DRM}(4)) - t_2(\text{DRM}(1))) = 5 - (7 - 2) + 5 - (5 - 2) = 2$$

Serim üzerinde en kısa yolun bulunması sonucunda elde edilen son istasyon atamaları aşağıdaki gibi hesaplanarak Tablo 3.7 'de özetlenmiştir.

$$\text{DRM}_8 - \text{DRM}_2 = \{1, 2, 3, 4, 5\} - \{1, 2\} = \{3, 4, 5\}$$

$$\text{DRM}_2 - \text{DRM}_0 = \{1, 2\} - \{\phi\} = \{1, 2\}$$

Tablo 3.7 Optimal İstasyon Atamaları

	1. Ürün		2. Ürün	
İstasyon	Görevler	Zamanı	Görevler	Zamanı
1	1, 2	5	1, 2	5
2	3, 4	4	4, 5	4

3.5.3. Modelin davranışları

Birleştirilmiş öncelik diyagramı nedeniyle birden fazla ürünlü montaj sistemi, tek bir ürün şeklinde ifade edilmiş, ve birkaç ilave özelliklerle, (Her bir ürün için çevrim zamanının geçilmemesi kontrolü, benzer görevlerin aynı istasyonda olması gibi) Gutjahr ve Nemhauser'in (1964) modeli birden fazla ürüne izin verecek şekilde düzenlenmiştir.

Modelimizde de, elde edilecek durum sayılarının, Gutjahr ve Nemhauser 'in (1964) sonuçlarıyla aynı olması beklenmektedir. Ancak ürün sayısı nedeniyle prosedürde, her bir ürün için çevrim zamanının geçilip geçilmeyeceği test edileceğinden, gerekli bilgisayar zamanları biraz daha fazla olacaktır. Halbuki Roberts ve Villa 'nın (1970) modelinde, kartezyen çarpımlar nedeniyle elde edilen durum sayıları oldukça fazladır. Ancak ürün sayısı 1 olduğu zaman Gutjahr ve Nemhauser 'in (1964) sonuçlarına ulaşılmaktadır.

Roberts ve Villa'nın (1970) modeli ile modelimiz arasında anlam bakımından farklılıklar vardır. Onların modelinde, her bir ürün ayrı ayrı değerlendirilmektedir. Ürünlerin ortak görevleri olsa dahi, birleştirilmiş diyagram dikkate alınmadığından, o görevler bile her ürün için ayrı ayrı görevlermiş gibi dikkate alınmaktadır. Modelimizde ise, ortak görevler ortak bir öncelik

diyagramında ifade edilir ve tek bir görev gibi algılanır. Ayrıca bu ortak görevler, doğal olarak da aynı istasyonlara atanırlar.

İki model arasındaki bu farklılığa rağmen, iki modelin küçük bir örnekle durum sayıları bakımından analizi, modelimizin sağladığı avantajın gösterilmesi açısından yararlı olabilir. Bu küçük analiz için, Bölüm 3.5.2 'deki örnek dikkate alınmıştır. Ele alınan problemde, ortak görevler mevcuttur. Ancak, Roberts ve Villa 'nın (1970) modeline göre bu ortak görevlerin ayrı ayrı dikkate alınması gerekmektedir. (Zira onların modelinde ortak görev kavramından hiç bahsedilmemiştir) elde edilen durum sayıları;

Roberts ve Villa 'nın modeli için : 20

Modelimiz için : 8 dir.

Eğer, problemdeki ürün sayısının 3 yada daha fazla olduğu göz önüne alınırsa, iki model arasındaki durum sayısı bakımından fark oldukça fazla olacaktır.

Tablo 3.8 'de, farklı E oranları ve farklı büyüklükteki problemler için gerekli durum sayıları verilmiştir.

Tablo 3.8 Modelin Durum Sayıları Bakımından Analizi

Görev Sayısı	E-Oranı	Öncelik İlişki Sayısı	Çözüm için Gerekli Durum Sayısı
11	0.45	13	69
16	0.49	21	277
30	0.11	34	87
30	0.44	33	1074
40	0.10	47	168

Tabloda, ürün adedi belirtilmemiştir. Zira ürün sayısı, çözüm için gerekli durum sayısını etkilememektedir. Tablo 3.8 'den de görüleceği gibi, durum sayısını artıran faktör görev sayısı değil, serimin E oranıdır. E oranı 1 'e yaklaştıkça, durum sayısının artışı büyümektedir.

Modelimiz, ürün sayısı kaç tane olursa olsun, kartezyen çarpımına gerek duymadan, tek bir ürün için hat dengeleniyormuş gibi durum sayısı gerektirmektedir. Her ne kadar tek ürünlü problemlerde dahi, E oranı 0.5 civarlarında olduğunda gerçek bir problemin (yaklaşık 70-100 görevli) En kısa yol algoritması ile çözümü bilgisayar zamanı bakımından çok güç olsa da, bu indirgeme çok ürünlü problemler için bir gelişme olarak kaydedilebilir. Ancak modelin yeni araştırmacılara ışık tutarak büyük ölçekli ve E oranı 0.5 'e yakın problemler için etkin hale getirilmesi ümit edilmektedir.

Bu bölümde, karışık modelli MHD problemleri için geliştirilen optimal metotlar detaylı olarak incelenmiş ve bu metotlara alternatif modeller önerilmiştir. Buna rağmen, MHD problemlerinin yapısı itibariyle, probleme optimal metotlarla çözüm aramak, özellikle problem boyutunun büyümesiyle birtakım zorlukları da beraberinde getireceğinden, bundan sonraki bölümde, aynı tür problemin çözümü için bir sezgisel metot geliştirilecektir.



BÖLÜM 4

KARIŞIK MODELLİ MONTAJ HATTI Dengelemesi için bir Sezgisel Yöntem

Karışık modellenli MHD problemler için geliştirilen optimal metotlardan bir tanesi (Bölüm 3) de incelenmiş ve alternatif bir 0-1 tamsayılı programlama modeli geliştirilmiştir. MHD problemlerinin yapısından dolayı, probleme optimal metotlarla çözüm aramak, özellikle problem boyutu büyüdüğünde bir takım zorlukları da beraberinde getirecektir. Bu nedenle bu tür problemlerin çözümünde sezgisel metotların kullanımı daha anlamlı ve gerçekçi olabilmektedir. Sezgisel metotlar, optimal çözümü garanti etmemekle birlikte, belirli kısıtlar altında, göreceli olarak iyi ve geçerli çözümleri daha az bir hesaplama ile sağlayabilmektedir.

Bu bölümde, karışık modellenli MHD problemleri için bir sezgisel metot geliştirilmiştir. Geliştirilen bu metot, Helgeson ve Birnie'nin (1961) tek modellenli hatlar için geliştirdiği RPW (Sıralı Pozisyonlu Ağırlık) tekniği ile benzerdir.

RPW tekniği, diğer alternatif yöntemlere göre daha çabuk kabul edilebilir iyi çözümler veren hızlı ancak yaklaşık bir metottur (Wild, 1989).

Geliştirilen sezgisel metodun avantajı, çok sade ve anlaşılır olmasıdır. Ayrıca nihai istasyon atamalarına erişimi de oldukça hızlıdır.

Sezgisel metotta, RPW tekniğinde olduğu gibi Pozisyon Ağırlıkları (PA) hesaplamaları vardır. Bu ağırlıklar, birleştirilmiş diyagrama göre hesaplanmalı ve büyükten küçüğe doğru sıralanmalıdır. RPW de tek modellenli için hesaplanan PA 'lar, karışık modellenli hatlarda kullanılabilecek şekilde düzenlenerek yeniden formüle edilmiştir.

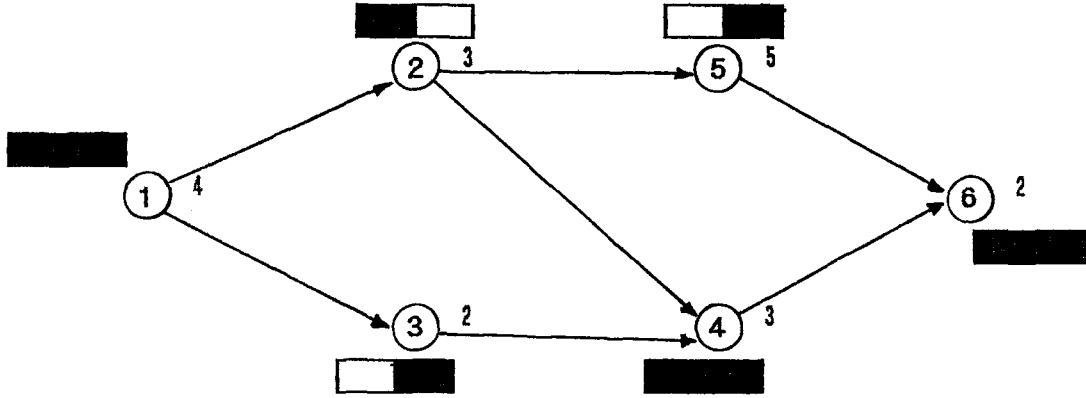
Bir karışık modellenli MHD problemi için PA hesaplaması, aşağıdaki gibi ifade edilmiştir:

$$PA_i = \sum_{m=1}^P t_{im} + \sum_{j \in S_1} \sum_{m=1}^P t_{jm} \quad i=1, \dots, N$$

Geliştirilen sezgisel metodun adımlarına geçmeden önce, birleştirilmiş diyagramda PA hesaplarının nasıl yapıldığı basit bir örnek üzerinde açıklanmıştır.

Örnek

Aynı üretim hattında üretilen iki ürüne ait birleştirilmiş öncelik diyagramı Şekil 4.1 'deki gibi oluşturulmuş olsun:



Şekil 4.1 İki Ürüne ait Birleştirilmiş Öncelik
Diyagramı

Düğümelerin yanındaki kutuların her bir hücresi bir modeli, taralı hücreler de o modelin o göreve sahip olduğunu (ya da o görevin ilgili modelin öncelik diyagramında var olduğunu) göstermektedir. Düğümelerin üzerindeki rakamlar, ilgili görevin görev zamanını ifade etmektedir. Ortak görevler için görev zamanları aynı olarak kabul edilmiştir.

Tablo 4.1 Öncelik ve Pozisyon Ağırlıkları Tablosu

Düğüm (Görev) No.	Görev Zamanı (dk)	1	2	3	4	5	6	Pozisyon Ağırlığı
1	4		*	*	+	+	+	28
2	3				*	*	+	18
3	2				*		+	12
4	3						*	10
5	5						*	9
6	2							4

Tablo 4.1 'deki * lar, bir görevin, hemen arkasından izleyen görev(ler)i, + lar ise, bir görevi diğer görevlerle olan ilişkileri nedeniyle izleyen görevleri göstermektedir. Tablo 4.1 'in son sütunundaki pozisyon ağırlıkları, her düğümün (görevin) kendi zamanı ile (ilgili düğüm birden fazla modele ait ise, o görevin zamanı; her bir ürün için ilgili görev zamanlarının toplamıdır.) bu görevi izleyen tüm görevlerin zamanlarının toplanmasıyla elde edilir. Örneğin, 1 nolu düğüm için PA değerleri şu şekilde hesaplanmıştır:

$$\begin{aligned}
PA &= 1.\text{görev} = 4+4=8 \\
&+ 2.\text{görev} = 3 \\
&+ 3.\text{görev} = 2 \\
&+ 4.\text{görev} = 3+3=6 \\
&+ 5.\text{görev} = 5 \\
&+ 6.\text{görev} = 2+2=4 \\
\text{Toplam} &= 28
\end{aligned}$$

Bu ağırlıklar, büyükten küçüğe doğru sıralanırlar. İstasyon atamaları yapılırken, enbüyük PA ya sahip olan göreve birinci önceliğe sahiptir.

4.1. Sezgisel Metodun Adımları

Adım 1. (Başlangıç değerleri)

Model(Ürün) sayısı=m; her bir modele ait çevrim zamanları CYC(m), $m=1, \dots, P$; Atanmamış görevler kümesi UASS= $\{1, 2, \dots, N\}$; Atanmış görevler kümesi ASS= $\{\phi\}$; İstasyona atanmaya aday görevler kümesi CND= $\{\phi\}$; İstasyon sayısı K=1; ve $RM(m)=CYC(m)$, $m=1, \dots, P$ dir.

Adım 2. Birleştirilmiş diyagramdan, ASS görev(ler)

kümesini dikkate alarak (eğer başlangıç ataması ise, kendisinden önce hiç görev olmayan düğümler CND listesine alınırırlar.) atanması uygun olan görevlerin bulunduğu CND listesi belirlenir. Eğer

CND listesi boş ise $CND=\{\phi\}$, işlem durdurulur.

Aksi durumda Adım 3'e gidilir.

Adım 3. CND listesindeki görevlerden PA sı en büyük olan i görevi seçilir ve Adım 4'e gidilir.

Adım 4. Eğer tüm modeller için $t_i(m) \leq RM(m)$ ise Adım 5'e, aksi durumda Adım 7'ye gidilir.

Adım 5. Model(ler)e ait i görevi $t_i(m)$ $m=1,..P$ zamanlarıyla k . istasyona atanır. Atanan i görevi UASS den çıkarılır. Yine atanan bu i görevi ASS kümesine ilave edilir. Ayrıca $RM(m)=RM(m)-t_i(m)$, $m=1,..P$ hesaplamaları yapılır ve Adım 6'ya gidilir.

Adım 6. Atanmamış görevler kümesinden (UASS), atanan görevler kümesinin (ASS) çıkarılmasıyla elde edilen NET kümesi boş ise (Atanacak başka görev kalmamışsa), işlem durdurulur. Aksi durumda Adım 2'ye gidilir.

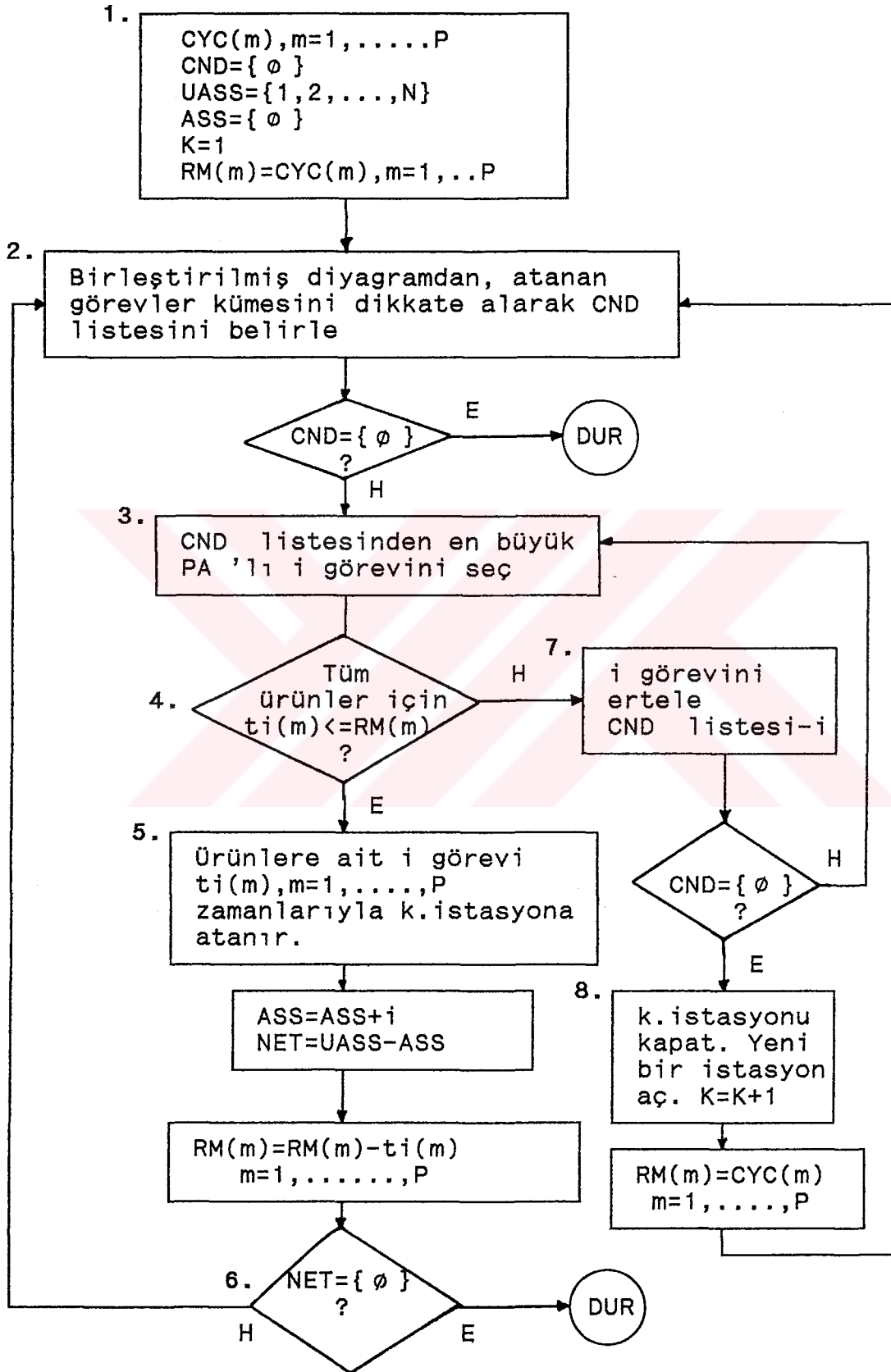
Adım 7. i görevi ertelenir ve i görevi CND listesinden (kümesinden) çıkarılır. Bu durumda CND listesi boş hale geliyorsa Adım 8'e, aksi durumda Adım 3'e gidilir.

Adım 8. k. istasyon kapatılır, yeni bir istasyon açılır.

K=K+1 ve $RM(m)=CYC(m)$, $m=1,...,P$ değer atamaları yapılır ve Adım 2'ye gidilir.

Şekil 4.2 'de, geliştirilen sezgisel metodun akış şeması verilmiştir.





Şekil 4.2 Geliştirilen sezgisel metodun akış şeması

4.2. Hesaplama Sonuçları

Karışık modelli MHD problemleri için geliştirilen sezgisel metot FORTRAN diliyle kodlanmış ve örnek bir için çözüm çıktılarıyla birlikte EK-3 'de verilmiştir. Metodun etkinliğinin araştırılması bakımından, farklı boyut, model ve E-Oranlarına sahip problemler için çalıştırılmıştır. Aynı şekilde, (Bölüm 3) de alternatif olarak geliştirilen ve optimal sonuç veren 0-1 tamsayılı programlama modeli de aynı problemler için çalıştırılmıştır. 0-1 tamsayılı programlama modelinin çözümü için GAMS [ver 2.25-1992] paket programı kullanılarak IBM 2155- 486DX2-66Mhz kişisel bilgisayarda yürütülmüş, çözücü (solver) olarak da OSL kullanılmıştır. Her bir problem için oluşturulan GAMS modellerinden birtanesi EK-4 'de verilmiştir.

Oluşturulan GAMS modelindeki tablo değerlerinin belirlenmesinde, FORTRAN diliyle kodlanan program WKM' nin çıktılarından (Bkz. EK-5) faydalanılmıştır.

Kullanılacak 0-1 tamsayılı programlama modelinin GAMS programına girdi olarak verilebilmesi için hazır hale getirilmesi, oldukça zaman almaktadır. Halbuki sezgisel metotta istenen girdi değerlerinin hazırlanması ve girilmesi hem çok kolay hem de çok hızlıdır. Oluşturulan MHD GAMS modeli üzerinde, problem özelliklerine göre kısıtları vs.), aynı modeli, kolaylık bakımından farklı

problem çözümleri için kullanmak mümkündür.

Tablo 4.2, Tablo 4.3, Tablo 4.4, Tablo 4.5, Tablo 4.6 da, sırasıyla 10, 20, 30, 40, 60 görevli, farklı model sayılı ve farklı E-Oranlı (0'a yakın, 0.5'e yakın ve 1'e yakın olmak üzere üç seviyeli) problemler için, Geliştirilen 0-1 modelin ve sezgisel metodun istasyon sayısı ve çözüm için gerekli bilgisayar zamanı açısından mukayeseleri, Tablo 4.7 de ise ilgili problemlerin çözümü için gerekli iterasyon sayıları ve mutlak tolerans sınırları verilmektedir.

Tablo 4.2 10 Görevli Problemler için, İki Metodun
İstasyon Sayısı ve Bilgisayar Zamanı Bakımından
Karşılaştırılması

Prob. No.	Ürün Sayısı	E-Oranı	Çevrim Zamanı	0-1 Formülasyon		Sezgisel Çözüm	
				K	Bilgisa. zam.(dk)	K	Bilgisa. zam.(sn)
1	3	0.644	11	3	0.269	4	0.05
2			6	8	48.53	8	0.05
3		0.444	11	3	0.015	3	0.05
4			6	8	20.81	8	0.05
5		0.113	11	3	0.009	3	0.05
6			6	8	6.800	8	0.05
7	2	0.644	16	2	0.012	2	0.05
8			9	4	0.075	4	0.05
9		0.444	16	2	0.013	2	0.05
10			9	4	0.026	4	0.05
11		0.113	16	2	0.011	2	0.05
12			9	4	0.015	4	0.05

Tablo 4.3 20 Görevli Problemler için, İki Metodun
İstasyon Sayısı ve Bilgisayar Zamanı Bakımından
Karşılaştırılması

Prob. No.	Ürün Sayısı	E-Oranı	Çevrim Zamanı	0-1 Formülasyon		Sezgisel Çözüm	
				K	Bilgisa. zam.(dk)	K	Bilgisa. zam.(sn)
13	4	0.71	10	7	14.657	8	0.17
14			28	3	33.146	3	0.17
15		0.536	10	7	13.678	8	0.16
16			28	3	3.251	3	0.17
17		0.147	10	8	9.433	8	0.27
18			28	3	0.104	3	0.27
19	3	0.71	17	4	32.399	4	0.16
20			13	5	27.250	5	0.16
21		0.536	17	4	12.139	4	0.16
22			13	5	12.328	6	0.16
23		0.147	17	4	0.073	4	0.27
24			13	6	15.304	6	0.27
25	2	0.71	20	3	15.124	4	0.17
26			9	7	26.249	7	0.16
27		0.536	20	3	2.537	4	0.16
28			9	7	22.437	8	0.16
29		0.147	20	4	0.344	4	0.27
30			9	8	6.976	8	0.27

Tablo 4.4 30 görevli Problemler için, iki Metodun
İstasyon Sayısı ve Bilgisayar Zamanı Bakımından
Karşılaştırılması

Prob. No.	Ürün Sayısı	E-Oranı	Çevrim Zamanı	0-1 Formülasyon		Sezgisel Çözüm	
				K	Bilgisa. zam.(dk)	K	Bilgisa. zam.(sn)
31	4	0.751	12	8	41.64	8	0.44
32			19	5	42.239	5	0.44
33		0.441	12	8	41.998	8	0.55
34			19	5	36.287	5	0.55
35		0.112	12	9	66.677	9	1.37
36			19	5	3.362	5	1.38
37	3	0.751	18	5	36.00	5	0.43
38			22	4	39.596	4	0.44
39		0.441	18	5	34.375	5	0.55
40			22	4	17.403	4	0.54
41		0.112	18	5	38.500	6	1.37
42			22	4	0.087	4	1.37
43	2	0.751	30	3	4.906	3	0.39
44			13	7	37.606	7	0.44
45		0.441	30	3	0.122	3	0.55
46			13	7	37.657	7	0.55
47		0.112	30	3	0.015	3	1.37
48			13	8	33.410	8	1.37

Tablo 4.5 40 Görevli Problemler için, iki Metodun
İstasyon Sayısı ve Bilgisayar Zamanı Bakımından
Karşılaştırılması

Prob. No.	Ürün Sayısı	E-Oranı	Çevrim Zamanı	0-1 Formülasyon		Sezgisel Çözüm	
				K	Bilgisa. zam.(dk)	K	Bilgisa. zam.(sn)
49	4	0.801	17	11	100.01	11	0.77
50		0.510	39	5	66.431	5	1.09
51		0.173	29	6	64.007	7	1.59
52	3	0.801	20	9	75.822	9	0.72
53		0.510	34	6	66.103	6	1.10
54		0.173	23	9	69.225	9	1.60
55	2	0.801	25	8	78.950	8	0.77
56		0.510	32	6	26.516	6	1.10
57		0.173	18	12	79.931	12	1.59

Tablo 4.6 60 Görevli Problemler için, iki Metodun
İstasyon Sayısı ve Bilgisayar Zamanı Bakımından
Karşılaştırılması

Prob. No.	Ürün Sayısı	E-Oranı	Çevrim Zamanı	0-1 Formülasyon		Sezgisel Çözüm	
				K	Bilgisa. zam.(dk)	K	Bilgisa. zam.(sn)
58	4	0.828	55	7	172.50	7	1.70
59		0.509	55	7	151.47	7	3.40
60		0.070	55	8	149.88	8	9.28
61	3	0.828	40	10	187.29	10	1.70
62		0.509	40	10	154.65	10	3.46
63		0.070	40	11	152.63	11	9.22
64	2	0.828	35	11	182.56	11	1.70
65		0.509	35	12	159.46	12	3.35
66		0.070	35	13	157.50	13	9.17

Tablo 4.7 10, 20, 30, 40 ve 60 Görevli Problemlerin Çözümü
için Gerekli İterasyon Sayıları ve Mutlak Tolerans
Sınırları

Prob. No.	İterasyon Sayısı	Tolerans Sınırı(%)	Prob. No.	İterasyon Sayısı	Tolerans Sınırı(%)
1	1699	0	21	40682	0
2	135669	12.2600	22	46947	10.2170
3	28	0	23	164	0
4	78439	0	24	50146	5.4400
5	1	0	25	60263	1.5409
6	30419	0	26	90000	8.3779
7	8	0	27	12304	0
8	216	0	28	98963	9.7440
9	16	0	29	1863	0
10	73	0	30	27377	0
11	2	0	31	126003	24.6324
12	18	0	32	150000	15.6860
13	49618	21.5888	33	150000	25.4202
14	90000	4.5000	34	148280	14.6921
15	50000	21.5436	35	223795	*
16	12181	0	36	13062	0
17	35546	21.4929	37	150000	10.1607
18	357	0	38	150000	2.5304
19	90000	3.2900	39	150000	7.9245
20	71700	7.4852	40	59923	0

Tablo 4.7 Devamı

Prob. No.	İterasyon Sayısı	Tolerans Sınırı(%)	Prob. No.	İterasyon Sayısı	Toler. Sınır. %
41	150000	7.6521	54	250000	*
42	262	0	55	250000	12.9172
43	16194	0	56	108525	6.6161
44	150000	8.6886	57	300000	*
45	556	0	58	450000	*
46	150000	8.1454	59	450000	*
47	21	0	60	450000	*
48	150000	9.8012	61	450000	*
49	248978	*	62	450000	*
50	250000	15.1252	63	450000	*
51	250000	16.3845	64	450000	*
52	250000	23.6811	65	450000	*
53	250000	6.6817	66	450000	*

* : ilgili iterasyonla halen tamsayılı sonuç alınamamıştır.

4.3. Sonuçların Değerlendirilmesi

0-1 formülasyonun ve sezgisel prodedürün, bilgisayar zamanı ve istasyon sayısı bakımından mukayeselerinin bulunduğu tablolardan görüleceği gibi, her iki metotdan elde edilen istasyon sayıları, büyük sayılabilecek boyuttaki problemler için dahi birbirlerine oldukça

yakındır. Sezgisel prosedür, bilgisayar zamanı bakımından da oldukça rahat görünmektedir. 0-1 formülasyonda, genellikle 40 görevli ve istasyon sayısı 9 dan küçük olan problemlere kadar çözüm alınabilmektedir. Çözüm elde ederken bazı problemler için, mutlak tolerans sınırları belirlenmiştir. Bunun anlamı, elde edilen değerin global optimumdan, tolerans sınırı kadar uzakta olması ya da o kadar sapması şeklindedir.

40 görevli bazı problemler için iterasyon sınırı 250000-300000 ve 60 görevli problemler için iterasyon sınırı 450000 verilmesine rağmen tamsayı bir netice elde edilemediğinden bu sınırdan işlem durdurulmuştur. Tablo 4.5 ve Tablo 4.6 daki $\frac{1}{K}$ ya karşılık gelen problemlere ait bilgisayar zamanları çözüm zamanlarını değil, modelin çalıştırılması işleminin durdurulmasına kadar geçen zamanı ifade etmektedir. İlgili iterasyonlarla tamsayı bir sonucun alınamadığı problemlerde, sezgisel prosedürden elde edilen değerlere dahi ulaşamamıştır. Eğer K değerinin başlangıçta sezgisel prosedürle değilde Robert ve Villa'nın 0-1 modelinde olduğu gibi (ürünler arasından en büyük eleman sayısına eşit) belirlendiği dikkate alınacak olursa, bu şartlar altında 10 görevli problemleri dahi çözmek imkansız hale gelecektir.

Netice itibariyle, geliştirilen sezgisel prosedürün hem optimal K değerlerine yakın olması ve hem de çok az bilgisayar zamanı gerektirmesi bakımından etkili olduğunu söylemek mümkündür.

BÖLÜM 5

KARIŞIK MODELLİ MHD PROBLEMLERİ İÇİN AMAÇ PROGRAMLAMA YAKLAŞIMI

MHD problemleri ile ilgili şu ana kadar yapılan çalışmaların tamamına yakın bir kısmında, sadece tek bir amacın dikkate alındığından söz edilmiştir. Modellerde genellikle ya çevrim zamanı, yada istasyon sayısı enküçüklenmeye çalışılmıştır. Bu iki amacın birbirleriyle çatışan amaçlar olduğuna da dikkat etmek gerekir (Yani, çevrim zamanı enküçüklenirken, istasyon sayısı artacak, istasyon sayısı enküçüklenirken de çevrim zamanı büyüyecektir).

Montaj hattı sistemlerinde, yukarıdaki amaçlara ek olarak birçok amaç tanımlamak da mümkündür: Bölgeleme kısıtlamaları, kapasite ekipman ve/veya işletim maliyetleri vs.

Şu ana kadar MHD problemlerinin çözümünde kullanılan tamsayılı programlama ve Amaç programlama metotlarının kullanım sıklıkları, Tablo 5 1 'de verilmiştir.

Tablo 5.1 Tamsayılı ve Amaç Programlama Metotlarının
Kullanım Sıklıkları

	Metotların Kullanım Sıklığı			
	TMD	TMS	ÇMD	ÇMS
Tamsayılı Programlama	7	0	1	0
Amaç Programlama	2	0	0	0

Tablo 5.1 'den görüldüğü gibi, TMD problemler için Gunther ve Arkadaşları (1983) ile Deckro ve Rangachari'ye (1990) ait iki çalışmadan başka Amaç programlamanın kullanıldığı başka bir problem grubu görülmemektedir. Halbuki Ghosh ve Gagnon (1989), "Yönetim birden fazla amaca erişmeyi isteyebilir. Bu nedenle montaj sistemlerinin modellenmesinde çok kriterli amaç fonksiyonlarının kullanımı, gerçek dünya amaçlarını yansıtması bakımından anlamlıdır ve bu konuda yapılacak olan araştırmalar ümit vericidir" diyerek bu konudaki araştırmacılara yol göstermiştir.

Yönetimin birden fazla amacı gerçekleştirme isteği dikkate alındığında, amaç programlama yaklaşımı bu tür bir modellemeye esneklik katacaktır.

Amaç programlama, belirli amaçlara mümkün olduğu kadar en iyi erişimin sağlanmasına yönelik girişimlerle

karakterize edilir. Bu erişim, hedef değerlerin altında kalınması ya da bu değerlerin geçilmesi şeklinde ölçülür. Amaç programlamada, amaç kısıtında belirtilen sağ taraf değerlerinden nümerik sapmaların enküçüklenmesi arzu edilir.

Amaç programlama metodu, esnek, etkin, gerçek problemlere uyumlu, geniş kullanım alanı olan ve model geliştirme ve çözüm metodu oldukça basit olan bir çok amaçlı karar verme tekniğidir (Ignizio, 1982)

Literatürdeki ÇMD problemler için böyle bir çalışmaya ihtiyaç duyulması nedeniyle bu problemler için bir 0-1 Amaç programlama modeli geliştirilmiştir. Modelin geliştirilmesinde, (Bölüm 3) de geliştirilen, kısıt ve değişken sayısı bakımından avantajlı 0-1 tamsayılı programlama modeli kullanılmış, amaç kısıtlarının oluşturulmasında Deckro ve Rangachari'nin (1990) modelinden faydalanılmıştır.

5.1. ÇMD Problemin Amaç Programlama Modeli

Bu bölümde sunulacak amaç programlama modelinin geliştirilmesinde, (Bölüm 3.2) de geliştirilen 0-1 model baz teşkil ettiğinden, o modelle ilgili tüm varsayımlar, hesaplamalar ve kısıtlar bu model için de geçerlidir. Bu nedenle bu bölümde, sadece üzerinde çalışılacak kısıtlar ve

yeni ilave kısıtlarla ilgili ifadeler yer alacaktır.

Atama Kısıtları

Atama kısıtları, genellikle amaç olarak dikkate alınmaya elverişli değildirler.

Öncelik Kısıtları

Bu kısıtlar, her bir modele ait öncelik diyagramlarına göre değil, birleştirilmiş öncelik diyagramına göre oluşturulurlar. Öncelik diyagramları, planlama aşamasında belirlendiğinden ve daha sonra görevlerin sıralamasında herhangi bir değişiklik arzu edilmediğinden, genellikle amaç olarak dikkate alınmazlar.

Çevrim Zamanı Kısıtları

Her hangi bir modelin görevlerinin atandığı her hangi bir istasyonun iş yükü, o modele ait çevrim zamanını geçemez. Çevrim zamanı; ya sabit bir değer, ya da alt ve üst sınır değerlerine sahip bir değişken olarak alınabilmektedir. (Bölüm 3.2) de hesaplanan E_i ve L_i değerleri, çevrim zamanının sabit olduğu durum dikkate alınarak hesaplanmıştır. Eğer çevrim zamanı aralık olarak ifade edilmiş ise, E_i ve L_i hesaplamalarında, C_m yerine $C_{müs}$ yazılarak sonuçlar alınmalıdır.

Çevrim zamanı sabit ise;

$$\sum_{i \in W_{km}} t_{im} \cdot V_{ik} \leq C_m \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, P \end{matrix}$$

Çevrim zamanı $C_{\min} \leq C_m \leq C_{\max}$ sınırlarına sahip bir değişken ise;

$$\sum_{i \in W_{km}} t_{im} \cdot V_{ik} - C_m \leq 0 \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, P \end{matrix}$$

Sabit ve değişken çevrim zamanları için amaç kısıtları, sırasıyla şu şekilde tanımlanabilir:

$$(1) \quad \sum_{i \in W_{km}} t_{im} \cdot V_{ik} + d_{km}^- - d_m^+ = C_m \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, P \end{matrix}$$

Buradaki sapma değişkeni d_m^+ , k kısıtların tümü için, her bir modele ilişkin istasyonlarda aynı çevrim zamanının olmasını temin etmek için kullanılmıştır.

$$(2) \quad \sum_{i \in W_{km}} t_{im} \cdot V_{ik} + d_{km}^- - C_m = 0 \quad \begin{matrix} k=1, \dots, K \\ m=1, \dots, P \end{matrix}$$

Yukarıdaki eşitlikte d_m^+ nın kaldırıldığını görüyoruz. Zira burada enküçüklenecek değer C_m dir.

Sabit ve değişken çevrim zamanı amaç kısıtları, çeşitli amaç imkanları sunmaktadır:

Bunlardan birincisi, d_{km}^- lerin toplamının enküçüklenmesidir ki bu, her bir modele ait her istasyondaki boş zamanları enküçükleyecektir.

Eğer bu, öncelikli amaç ise, d_m^+ , (1) kısıtından kaldırılmalıdır. (2) nolu kısıtta ise, her istasyondaki her bir modele ait boş zamanlar, çevrim zamanıyla ödünleştirecektir.

ikinci durum, istasyonlar arasındaki mücadele edilen sapmaların sınırlamaları ile ilgilidir.

(1). kısıt tipi için, d_m^+ ile ilgili bir üst sınırın belirlenmesi, çevrim zamanındaki muhtemel bir artışı sınırlayacaktır. d_{km}^- ile ilgili bir üst sınır ise, her iki tip çevrim zamanı kısıtları için, her hangi bir modele ait istasyondaki mücadele edilebilir boş zamanı sınırlayacaktır. Bu sınırlama kısıtları şu şekilde ifade edilebilir;

$$\begin{array}{ll} i) & d_m^+ \leq US \quad \begin{matrix} m=1, \dots, P \\ k=1, \dots, K \end{matrix} \\ ii) & d_{km}^- \leq USN \end{array}$$

Yukarıdaki sınırlama kısıtları, kendi kendilerine amaç olarak da ifade edilebilirler.

Bölgeleme Kısıtları

i) Uyumsuz bölgeleme kısıtları

Bu kısıtlar, belirli bir istasyonda, birlikte olması istenmeyen görevlerin, aynı istasyonda yapılmalarını engeller.

a 'nın b 'den sonra geldiği iki görev için aşağıdaki bölgeleme kısıtı yazılabilir:

$$\sum_{k=E_a}^{L_a} k.V_{ak} - \sum_{k=E_b}^{L_b} k.V_{bk} \geq 1$$

Bu uyumsuz bölgeleme kısıtı, sapma değişkenlerin eklenmesiyle, bir amaç kısıtı olarak ifade edilebilir.

$$\sum_{k=E_a}^{L_a} k.V_{ak} - \sum_{k=E_b}^{L_b} k.V_{bk} + d_n^- - d_n^+ = 1$$

Eğer $d_n^- = 0$ ise, uyumsuz a ve b görevleri aynı istasyona atanmayacaklardır.

ii) Uyumlu bölgeleme kısıtları

Uyumlu bölgeleme, uyumsuz bölgeleme kısıtının basit bir varyasyonudur. Kısıt, belirli bir istasyonda, belirli bir modele ait birlikte olması istenen görevlerin, aynı istasyonda yapılmalarını (ortak parça, techizat kullanım vs den dolayı) temin eder.

$$\sum_{k \in E_a}^{L_a} k \cdot V_{ak} - \sum_{k \in E_b}^{L_b} k \cdot V_{bk} = 0$$

Kısıt, sapma değişkenlerin ilavesiyle, bir amaç kısıtı olarak düzenlenebilir.

$$\sum_{k \in E_a}^{L_a} k \cdot V_{ak} - \sum_{k \in E_b}^{L_b} k \cdot V_{bk} + d_c^- - d_c^+ = 0$$

Eğer d_c^- ve d_c^+ sıfıra enküçüklenirse, uyumlu a ve b görevleri, aynı istasyona atanacaklardır. a ve b 'nin aynı istasyona atanmaları mümkün değilse, o zaman diferansiyel (nümerik) ağırlıkların kullanımıyla a ya da b görevlerinden hangisinin istasyona ilk olarak atanacağı konusunda bir öncelik tanımlanabilir. Uyumlu atamalar yapılamadığında, bu ağırlıklar ilave maliyet gerektirebilirler.

İstasyon Kısıtları

ÇMD problemlerde, her bir model için istasyon sayısının aynı olması, düzenli akışın temini, boş zamanın azaltılması bakımından oldukça önemlidir.

(Bölüm 3.2) 'deki

$$\sum_{m=1}^P X_{km} - P \cdot A_k = 0 \quad k=1, \dots, K$$

yardımcı kısıtı, istasyon sayısının tüm modeller için aynı olmasını temin eder.

Eğer belirli bir istasyon sayısı arzu ediliyorsa (S), o zaman aşağıdaki kısıtı üretebiliriz.

$$\sum_{k=1}^K A_k \leq S$$

Kısıt, sapma değişkenlerinin eklenmesiyle amaç kısıtına dönüştürülebilir;

$$\sum_{k=1}^K A_k + d^- - d^+ = S$$

Kısıt, model sayısı ne kadar olursa olsun tek satırdır.

Eğer d^+ sifıra enazlanabilirse, kullanılan istasyon sayısı S ya da daha küçük olacaktır. d^- nin enbüyüklenmesi, kullanılan toplam istasyon sayısının enküçüklenmesine etkide bulunacaktır.

İstasyon sayısının, çevrim zamanı kısıtında olduğu gibi aralık şeklinde belirlenmesi mümkünse, istasyon kısıtını da o şekilde düzenlemek mümkündür.

İstasyon ile ilgili diğer bir kısıt da, bütçeleme

kısıtıdır. B , istasyonların kurulması için planlanmış toplam bütçe ve b_k da k istasyonunun maliyeti olmak üzere, aşağıdaki bütçeleme kısıtı yazılabilir;

$$\sum_{k=1}^K b_k \cdot A_k \leq B$$

Bütünleşik planlama aşamasında, bu bütçe kısıtı, bir amaç kısıtı olarak ifade edilebilir;

$$\sum_{k=1}^K b_k \cdot A_k + d_B^- - d_B^+ = B$$

d_B^- ve d_B^+ , hedeflenmiş bütçeden sapmaları vermektedir.

Montaj hatlarıyla ilgili bu tür amaçları genişletmek mümkündür. Ancak sistemdeki bir problemi tanımlarken, gereksiz kısıtlamalara gidilmemesi fayda sağlayacaktır. Zira fazla değişken ve kısıtlardan oluşan bir problemin çözümü gittikçe karmaşık bir hale gelecektir.

Amaçlar

Amaçlar, arzu edilen amaç seviyelerinden, ağırlıklı sapmaları enküçükleyecek şekilde oluşturulurlar. Amaç programlamada, amaç fonksiyonundaki seçilmiş ağırlıklar, belirli amaçların öncelik durumlarını ifade etmektedirler. Bu ağırlıklar nümerik ya da öncelikli (preemptive) olarak seçilebilirler; Nümerik ağırlıklama. aynı öncelik

seviyesindeki amaçlar arasında farklılaştırma yapmak için kullanılırlar. Öncelikli ağırlıklamada, düşük öncelikli amaçlar dikkate alınmadan önce, yüksek öncelikli amaçların sağlanmasına çalışılır. Çalışmanın bu kısmında, öncelikli ağırlıklama yöntemi kullanılacaktır.

P_s , s önceliğine atanan öncelikli ağırlığı göstermek üzere $P_s \gg P_{s+1}$ şeklinde tanımlanabilir. Önceliklerin sıraları değiştirilmek suretiyle, öncelik yapısının duyarlılığı araştırılabilir.

Hat dengeleme problemlerinde, problemin yapısına bağlı olarak birçok amaç geliştirmek mümkündür. Örneğin, Gunther ve Arkadaşları (1983) otomobil üretimi ile ilgili, uyumlu ve uyumsuz bölgeleme tipi, çevrim zamanının geçilmemesi, istasyonlardaki görev sayılarının ayarlanması, istasyon sayısının enazlanması vb. amaçları içeren 11 amaç belirlemişlerdir.

Amaç programlamada, amaç fonksiyonunun öncelikli ağırlıklı tanımlanmasıyla ilgili basit bir örnek verilmiştir.

Örnek

Bir montaj hattında, P_1 amacı P_2 amacından öncelikli olmak üzere ($P_1 > P_2$) iki amaç belirlenmiş olsun.

1. Amaç: P_1 : Çevrim zamanı belirli bir zamandan fazla olmasın.

2. Amaç: P_2 : İstasyon sayısı belirli bir sayıdan fazla olmasın.

Öncelikli amaç fonksiyonu (kısıtı) şu şekilde yazılabilir:

$$\text{Min } Z : P_1(CYC1+CYC2), P_2(STA)$$

CYC1,CYC2 ve STA sırasıyla, 1.ve 2. ürün için çevrim zamanı ve istasyon amaç kısıtlarının pozitif sapma değişkenleridir.

Öncelikli amaç programlamaya göre, 1. amaç daha büyük önceliğe sahip olduğundan, problem önce 1. amacı (P_1) sağlamak üzere çözülecektir. Daha sonra 2. öncelikli amacın (P_2) tatminine çalışılacaktır. 1. öncelikli amaçtan elde edilen sonuç (sapma değişken değeri), ikinci öncelikli amaca kısıt olarak ilave edilecektir. Dolayısıyla, 2. öncelikten elde edilen çözüm, 1. amacı da dikkate almış olacaktır.

Şimdi, ÇMD problemler için geliştirilen 0-1 amaç programlama modeli, örnek problem(ler) üzerinde denenecek ve önceliklerin sıraları değiştirilmek suretiyle, öncelik yapısının duyarlılığı araştırılacaktır.

5.2. Modelin Örnek Problemlerle Denenmesi ve Öncelik

Yapılarının Duyarlılığının Araştırılması

Model, tek amaçlı 0-1 modeli için (Bkz. bölüm 3.2) gerekli bütün 0-1 değişkenlerine sahiptir. Ancak, amaçlar için gerekli sapma değişkenleri modele ilave edilmişlerdir.

Amaç programlamada çözüm, genelde şu iki metotla gerçekleştirilir:

1- Sıralı Amaç Programlama.

2-Çok safhalı veya Geliştirilmiş Simpleks Metodu.

Çok safhalı simpleks' de, amaçların tümü aynı anda sağlanmaya çalışıldığından, bilgisayar zamanı oldukça büyüktür. Halbuki sıralı amaç programlamada, her bir öncelik için ayrı ayrı çözüm yapıldığından, bilgisayar zamanı çok safhalı simpleks'e nazaran oldukça küçüktür. Ayrıca, Sıralı amaç programlama'nın ayrı ayrı çözüm yapması nedeniyle, geliştirilen tek amaçlı ticari doğrusal programlama ya da tamsayılı programlama yazılımlarını kullanma şansı da vardır (Ignizio, 1982).

Bu nedenle, geliştirilen 0-1 amaç programlama modelinin çözümü için sıralı amaç programlama metodu kullanılacaktır.

Sıralı amaç programlama için gerekli bilgisayar zamanının, 0-1 model için (Bkz. Bölüm 3.2) gerekli bilgisayar zamanından daha az ya da en fazla ona eşit olduğu bilinmektedir. Zira, sıralı amaç programlamada bir amaç tatmin edilebilirse, optimaliteyi kurmak için bilgisayar zamanı harcamak gereksizdir. Amaçtaki sapma değişkenleri sıfıra enküçüklenirse, amaç için ayrı bir hesaplama yapılmasına gerek yoktur (Deckro and Rangachari, 1990).

Bütün bu nedenlerden dolayı, 0-1 tamsayılı programlama modeli (Bkz. Bölüm 3.2) önceki bölümlerde, farklı problem durumları için detaylı olarak analiz edildiğinden, 0-1 amaç programlama modelinin farklı problemler için denenmesine gerek duyulmamaktadır. Sadece model, öncelik yapılarının duyarlılığının araştırılması bakımından, örnek bir problem üzerinde işletilecektir.

Örnek problem olarak (Bölüm 3.3)' deki iki ürünlü problem seçilmiştir. 0-1 amaç programlamayla modellenen problemde, öncelik durumuna göre her bir model sıralı olarak LINDO [1988] paket programıyla çalıştırılmıştır. Problemde, karar verici tarafından 3 amacın belirlendiği ve bu amaçların P_1 , P_2 , P_3 önceliğinde olduğu varsayılmıştır. Bu amaçlar aşağıda sıralanmıştır:

1. Öncelikli Amaç: P_1 : İş istasyon sayısı 3 ten fazla olmamalıdır.
2. Öncelikli Amaç: P_2 : Çevrim zamanı, 1.ürün için 10 dk'yı 2. ürün için 8 dk' yı geçmemelidir.
3. Öncelikli Amaç: P_3 : 1 ve 9 nolu görevlerin (her iki görev de iki ürün için ortak) aynı istasyonda yapılmasına izin verilmemelidir.

STA, istasyon amaç kısıtının pozitif sapma değişkeni, CYC1 ve CYC2, sırasıyla 1. ve 2. ürün için çevrim zamanı amaç kısıtlarının pozitif sapma sapma değişkenlerini ve BOLGE de, bölgeleme amaç kısıtının negatif sapma değişkenini temsil etmek üzere, sıralı olarak amaç fonksiyonu şu şekilde ifade edilebilir:

$$\text{Min } P_1(STA), P_2(CYC1+CYC2), P_3(BOLGE)$$

$P_1 > P_2 > P_3$ 'den , en yüksek önceliğe sahip P_1 için model çalıştırılır.

$$\text{Min } P_1(STA)$$

Çözüm sonucunda elde edilen STA değeri (a^* olsun), 2. öncelikle çalıştırılacak modele kısıt olarak ilave edilir.

$$\begin{array}{ll} \text{Min } P_2(CYC1+CYC2) \\ \text{st.} \end{array}$$

$$STA = a^*$$

Daha sonra aynı prosedür P_3 için de tekrarlanır (Ignizio,1982). P_1 , P_2 , ve P_3 için geliştirilen 0-1 amaç programlama modelleri ve örnek problem için herbirinin çözüm çıktıları, sırasıyla EK-6, EK-7, EK-8 de verilmiştir.

Örnek problem için, önceliklerin duyarlılıkları araştırılmış ve Tablo 5.2 'de verilmiştir.

Tablo 5.2 Önceliklerin Duyarlılık Analizi

prob. No.	1. Öncelik	2. Öncelik	3. Öncelik	Saglanamayan Öncelik/Amaç ^a	a 'in sapma Deg. değeri	toplam Çevrim Z.	İstas. Sayısı
1	P_1	P_2	P_3	$2,3/P_2, P_3$	CYC1+CYC2=2 BOLGE=1	20	3
2	P_2	P_1	P_3	$2/P_1$	STA=1	18	4
3	P_2	P_3	P_1	$3/P_1$	STA=1	18	4
4	P_1	P_3	P_2	$3/P_2$	CYC1+CYC2=3	21	3
5	P_3	P_2	P_1	$3/P_1$	STA=1	18	4
6	P_3	P_1	P_2	$3/P_2$	CYC1+CYC2=3	21	3

Aynı analiz için, her bir ürüne ait çevrim zamanlarının ayrı ayrı ifade edilmesi, karar vericinin ürün bazında karar vermesine ışık tutması ve kararın kolaylaşması bakımından faydalı olabilir. Bu şekildeki ifade Tablo 5.3 'de verilmiştir.

Tablo 5.3 Önceliklerin Duyarlılık Analizi (Detaylı
Çevrim zamanlı)

Prob. No	1. Öncelik	2. Öncelik	3. Öncelik	Sağlanamayan Öncelik/Amaç ^a	a 'nın sapma Değ.Değeri	Çevrim Za.		İst. Say.
						1.Ürün	2.Ü	
1	P ₁	P ₂	P ₃	2,3/P ₂ ,P ₃	CYC1=0,CYC2=2 BOLGE=1	10	10	3
2	P ₂	P ₁	P ₃	2/P ₁	STA=1	10	8	4
3	P ₂	P ₃	P ₁	3/P ₁	STA=1	10	8	4
4	P	P	P	3/P	CYC1=0,CYC2=3	10	11	3
5	P ₃	P ₂	P ₁	3/P ₁	STA=1	10	8	4
6	P ₃	P ₁	P ₂	3/P ₂	CYC1=0,CYC2=3	10	11	3

Tablo 5.2 ve Tablo 5.3 'den görüldüğü gibi, bütün öncelik sıralamalarında, 3. amacı aynı anda sağlamak mümkün olmamıştır. 3. önceliğin (P₃), 1. problem dışında diğer bütün sıralamalarda sağlandığı açıkça görülmektedir. Bunun dışında, ya P₂, ya da P₁ öncelikli amacı, arzu edilen çözümden sapmaya neden olmuşlardır.

Yukarıdaki analizde, karar vericiye amaçlarla ilgili 3! kadar geniş bir tercih imkanı sağlanmıştır. Karar verici, son seçimi kendi tercihinine göre yapacaktır. Yani, ya çevrim zamanındaki, ya da istasyon sayısındaki sapmalardan birisine göz yumarak tercihinin yapacaktır.

SONUÇ VE TARTIŞMA

Bu çalışmada, MHD problemlerinin bir alt sınıfı olan Karışık modelli problemlerle ilgili metotlar geliştirilmiştir.

İlk olarak, Roberts ve Villa tarafından geliştirilen 0-1 Tamsayılı programlama modeli incelenmiş ve yeni bir alternatif 0-1 Tamsayılı programlama modeli geliştirilmiştir. Geliştirilen model, kısıt ve değişken sayısı bakımından Roberts ve Villa' nın modeline göre daha avantajlıdır. Kısıt ve değişken sayısının azalmasını şu ilave özellikler temin etmiştir;

- i) Her bir ürüne ait öncelik diyagramlarının tek bir diyagram halinde birleştirilmesi
- ii) Başlangıçta belirlenen K 'nın (en fazla mücade edilen istasyon sayısı) bir sezgisel metotla tesbit edilmesi
- iii) En erken istasyon (E_i) ve En geç istasyon (L_i) hesaplamaları
- iv) Farklı ürünlere ait ortak görevlerin aynı istasyona atanmaları gerektiği varsayımı.

Geliştirilen 0-1 Tamsayılı programlama modeli, farklı boyuttaki problemler için denenmiş ve sonuçlar değerlendirilmiştir.

ikinci olarak yine Roberts ve Villa tarafından geliştirilen En kısa yol modeli incelenmiş ve alternatif bir serim modeli önerilmiştir. Önerilen serim modeli, Gutjahr ve Nemhauser 'in TMD problemler için geliştirdiği En kısa yol modeline dayanmaktadır. Yukarıdaki 1 ve 4 nolu özellikler, önerilen model için de kullanılmıştır. Roberts ve Villa'nın modelinde, ürünlere ait durumlar belirlenip bunların kartezyen çarpımı alındığından, çözüm için gerekli durum sayısı çok fazladır. Ürün sayısının 3 ya da daha fazla olduğu orta büyüklükteki bir problem için elde edilen durum sayısı, çözümü hemen hemen imkansız hale getirmektedir. Modelimiz ise, birleştirilmiş diyagram kullanıldığından, ürün sayısı kaç tane olursa olsun tek bir ürün için hat dengeleniyormuş gibi durum sayısı gerektirmekte ve bu da çözümü kolaylaştırmaktadır. Önerilen bu modelin davranışları incelenmiş ve farklı görev ve E Oranlarına sahip problemler için durum sayıları bakımından analizi yapılmıştır.

MHD problemlerinin yapısından dolayı, optimal metotlarla çözüm, özellikle problem boyutu büyüdüğünde bir takım zorlukları da beraberinde getirmektedir. Bu nedenle üçüncü olarak, karışık modelleri MHD problemleri için bir sezgisel metot sunulmuştur. Sunulan bu metot, Helgeson ve Birnie 'nin Tek modelleri hatlar için geliştirdiği RPW tekniğine dayanmaktadır. Sezgisel metot, 10, 20, 30, 40 ve 60 görevli farklı çevrim zamanı ve E-Oranlarına sahip

problemler için, Robert ve Villa'nın modeline alternatif olarak geliştirilen 0-1 tamsayılı programlama modeliyle istasyon sayıları ve bilgisayar zamanı bakımından karşılaştırılmıştır. Elde edilen K değerleri, büyük sayılabilecek boyuttaki problemler için dahi birbirlerine oldukça yakın bulunmuştur. 0-1 Tamsayılı programlama formülasyonunda, genellikle 40 görevli ve istasyon sayısı 9 dan küçük olan problemlere kadar çözüm alınabilmektedir. 60 görevli problemlerde, iterasyon sınırı 450000 verilmesine rağmen tamsayılı bir sonuç elde edilemediğinden bu sınırda işlem durdurulmuştur. 450000 iterasyondaki her bir problem, K değeri sezgisel metotla belirlenmesine rağmen yaklaşık 3 saatlik bir zaman almıştır. Sezgisel metodun ise, hem optimal K değerlerine yakınlık ve hem de çok az bilgisayar zamanı gerektirmesi bakımından etkili olduğu gözlenmiştir.

Son olarak, yöneticiye birden fazla amacı gerçekleştirme imkanı vermesi ve literatürde şu ana kadar hiç rastlanmaması bakımından, karışık modellenli MHD problemleri için Amaç programlama yaklaşımı kullanılarak yeni bir 0-1 Amaç programlama modeli geliştirilmiştir. Geliştirilen model, örnek problemlerle denenmiş ve öncelik yapılarının duyarlılığı araştırılmıştır. Amaç programlamada çözüm metodu olarak da sıralı amaç programlama kullanılmıştır.

ÇMD MHD problemleriyle ilgili yapılan bu yeni

arařtırmalarda ama, bu problemlerle ilgili halen ok az olan alıřmalara yenilerinin eklenip literatre katkı saėlayabilmek ve bu konuda alıřacak olan arařtırmacılarla yeni bir bakıř aısı sunabilmektedir.

ileride yapılması dřnlen alıřmalar ařaėıda sıralanmıřtır :

i) Geliřtirilen modellerin, gerek bir montaj hattı sistemindeki problemlerin zm iin alıřtırılması (uygulama alıřması)

ii) Farklı E-Oranlı ve boyutlu test problemlerinin retilmesini saėlayan bir yazılımın oluřturulması

iii) Model sıralama metotlarını da dikkate alarak dengelemenin daha etkin bir hale getirilmesi

iv) Blm 3 'de geliřtirilen serim modeli iin bilgisayar yazılımının oluřturularak, farklı boyuttaki problemler iin davranıřlarının incelenmesi

v) Kombinatoryal optimizasyon problemleri iin kullanılan Genetik Algoritmalarının probleme uygulanması ve davranıřlarının incelenmesi

KAYNAKLAR

- Acar, N., Estaş, S., " Kesikli Seri Üretim Sistemlerinde Planlama ve Kontrol Çalışmaları",MPM Yayınları no:309 Ankara,1986.
- Agrawal, P.K., "The related activity concept in assembly line balancing", International Journal of Production Research (IJPR), vol.23, no.2, 1985.
- Akagi, F., Osaki, H., Kikuchi, S. "A method for assembly line balancing with more than one worker in each station", IJPR, vol.21, no.5, 1983.
- Arcus, A.L., " COMSOAL a Computer method of sequencing operations for assembly lines",IJPR, vol.4, no.4, 1966.
- Bhattacharjee, T.K., Sahu, S., " Complexity of single model assembly line balancing problem", Engineering Cost and Production Economics, vol.18, 1990.
- Bowman, E.H., " Assembly line balancing by linear programming", Operations Research (OR), vol.8, no.3, 1960.
- Buxey, G., " Incompletion costs versus labor efficiency on the fixed-item moving belt flowline",IJPR, vol.16, no.3, 1978.
- Chakravarty, A.K., Shtub, A., " Balancing mixed model lines with in-process inventories", Management Science (Man.Sci), vol.31, no.9, 1985.

- Chakravarty, A.K., Shtub, A., " The effect of learning on the operation of mixed model assembly lines", Production and Operations Management, vol.1, no.2, Spring,1992.
- Dar-El, E.M., " MALB-A heuristic technique for balancing large single-model assembly lines",AIIE Transactions, vol.5, no.4, 1973.
- Dar-El, E.M., " Mixed-model assembly line sequencing problems", OMEGA, vol.6, no.4, 1978.
- Dar-El, E.M., Cothor, R.F., " Assembly line sequencing for model mix",IJPR, vol.13, no.5, 1975.
- Dar-El, E.M., Rubinovitch, Y., " MUST-A multiple solutions technique for balancing single model assembly lines" ,Man.Sci, vol.25, no.11, 1979.
- Dar-El, E.M., Rubinovitch, Y., " Using learning theory in assembly lines for new products", International Journal of Production Economics, vol.25, 1991.
- Deckro, R.F., " Balancing Cycle time and Workstations",IIE Transactions, vol.21, no.2, 1989.
- Deckro, R.F., Rangachari, S., " A Goal Approach to Assembly Line Balancing",Computers Operations Research, vol.17, no.5, 1990.
- Dervitsiotis, K.N., " Operations management", McGraw-Hill Book Company, 1981
- Driscoll, J., Abdel-Shafi, A.A.A., " A Simulation approach to evaluating assembly line balancing solutions "

- ,IJPR, vol.23, no.5, 1985.
- El-Sayed, E.A., A.Raouf., Tsui, C.L., " A new heuristic approach to assembly line balancing",Computers & Industrial Engineering, vol.4, no.3, 1980.
- Erel, E., " A Methodology to solve single-model assembly line balancing problem and its extention", Ph.D Thesis, Blacksburg, Virginia, may 1987.
- Erel, E., " Stokastik montaj hattı dengeleme problemi üzerine yapılan araştırmalar", Endüstri Mühendisligi Dergisi, yıl.3, sayı.13, 1991.
- Ghosh, S., R. Gagnon, J., " A Comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems",IJPR, vol.27, no.4, 1989.
- Gutjahr, A.L., Nemhauser, G.L., " An algorithm for the line balancing problem",Man.Sci, vol.11, no.2, 1964.
- Gunter, R.E., Johnson, G.D., Peterson, R.S., " Currently practized formulations for the assembly line balance problem", Journal of Operations Management, vol.3, no.4 1983.
- Held, M., Karp, R.M., " A Dynamic Programming approach to sequencing problem", Journal of the Society of Industrial and Applied Math.,vol.10, no.1, 1962.
- Held, M., Karp, R.M., Shreshian, R., " Assembly line balancing: Dynamic programming with precedence constraints", OR, vol.11, no.3, 1963.

- Helgeson, W.B., Birnie, D.P., " Assembly line balancing using the ranked positions1 weight technique", *Journal of Industrial Engineering*, vol.12, no.6, 1961.
- Hoffmann, T.R., " Assembly line balancing with a precedence matrix", *Man. Sci.*, vol.9, no.4, 1963.
- Hu, T.C., " Parallel sequencing and assembly line problems", *OR*, vol.9, Nov-Dec.,1961.
- Ignall, E.J., " A Review of assembly line balancing", *Journal of Industrial Engineering*, vol.16, no.4, 1965.
- Ignizio, J.P., "Linear programming in single and multiple objective systems", Prentice-Hall, N.J., 1982
- Jackson, J.R., " A Computing procedure for a line balancing problem", *Man.Sci.*, vol.2, no.3, 1956.
- Johnson, R.V., " Assembly line balancing algorithms: Computation comparisons", *IJPR*, vol.19, no.3, 1981.
- Johnson, R.V., " A Branch and Bound Algorithm for assembly line balancing problems with formulation irregularities", *Man.Sci*, vol.29, no.11, 1983.
- Kao, E.P.C., " A Preference order dynamic program for stochastic assembly line balancing", *Man.Sci.*, vol.22, no.10, 1976.
- Kao, E.P.C., " Computational experience with a stochastic assembly line balancing algorithm", *Computers & Operations Research*, vol.6, no.2, 1979.

- Kao, E.P.C., Queyranne, M., " On dynamic programming methods for assembly line balancing", *OR*, vol.30, no.2, 1982.
- Kilbridge, M.D., Wester, L., " A heuristic method of assembly line balancing", *J. of Industrial Engineering*, vol.12, no.4, 1961.
- Klein, M., " On assembly line balancing", *OR*, vol.11, no.2, 1963.
- Kottas, J.F., Lau, H.S., " A Cost oriented approach to stochastic line balancing", *AIIE Transactions*, vol.5, no.2, 1973.
- Kottas, J.F., Lau, H.S., " A total operating cost model for paced lines with stochastic task times", *AIIE Transactions*, vol.8, no.2, 1976.
- Kottas, J.F., Lau, H.S., " A stochastic line balancing procedure", *IJPR*, vol.19, no.2, 1981.
- Lehman, M., " On criteria for assigning models to assembly lines", *IJPR*, vol.7, no.4, 1969.
- Macaskill, J.L.C., " Production line balances for mixed-model lines", *Man.Sci.*, vol.19, no.4, 1972.
- Macaskill, J.L.C., " Computer Simulation for mixed-model production lines", *Man.Sci.*, vol.20, no.3, 1973.
- Malakooti, B., " A multiple criteria decision making approach for the assembly line balancing problem", *IJPR*, vol.29, no.10, 1991.
- Mansoor, E.M., Ben-Tuvia, S., " Optimizing balanced assembly lines", *Journal of Industrial Engineering*,

- vol.17, no.3, 1966.
- Master, A.A., " An experimental investigation and comparative evaluation of production line balancing techniques", *Man.Sci.*, vol.16, no.11, 1970.
- Moberly, L.E., Wyman, P.F., " An application of simulation to comparison of assembly line configurations, *Decision Sciences*, vol.4, october, 1973.
- Moodie, C.L., Young, H.H., " A heuristic method of assembly line balancing for assumptions of constant or variable work element times", *Journal of Industrial Engineering*, vol.16, no.1, 1965.
- Nevins, A.J., " Assembly line balancing using best bud search", *Man. Sci.*, vol.18, no.9, 1972.
- Patterson, J.H., Albracht, J.J., " Assembly line balancing: 0-1 programming with fibonacci search", *OR*, vol.23, no.1, 1975.
- Pinto, P.A., Dannenbring, D.G., Khumawala, B.M., " A branch & bound algorithm for assembly line balancing with paralleling", *IJPR*, vol.13, no.2, 1975.
- Pinto, P.A., Dannenbring, D.G., Khumawala, B.M., " A heuristic network procedure for the assembly line balancing problem", *Naval Research Logistic Review*, vol.25, no.2, 1978.
- Reeve, N.R., Thomas, W.H., " Balancing stochastic assembly lines", *AIIE Transactions*, vol.5, no.3, 1973.
- Roberts, S.D., Villa.C.D., " On a multiproduct assembly

- line balancing problem", *AIIE Transactions*, vol.II, no.4, 1970.
- Salveson, M.E., " The assembly line balancing problem", *The Journal of Industrial Engineering*, vol.6, no.3, 1955.
- Sarin, S.C., Erel, E., " Development of cost model for the single-model stochastic assembly line balancing problem", *IJPR*, vol.28, no.7, 1990.
- Schrage, L., Baker, K.R., " Dynamic programming solution of sequencing problems with precedence constraints, *OR*, vol.26, may-june, 1978.
- Sculli, D., " Dynamic aspects of line balancing", *OMEGA*, vol.7, no.6, 1979.
- Sculli, D., " Short term adjustment to production lines", *Computers & Industrial Engineering*, vol.8, no.1, 1984.
- Shin, D., " An efficient heuristic for solving stochastic assembly line balancing problem", *Computers & Industrial Engineering*, vol.18, no.3, 1990.
- Shin, D., Hokey, M., " Flexible line balancing practices in a Just-In-Time environment", *Production & Inventory Management*, fourth Quarter, 1991.
- Shtub, A., " The effect of incompleteness cost on the line balancing with multiple manning of work station", *IJPR*, vol.22, no.2, 1984.
- Silverman, F.N., Carter, J.C., " A Cost-based methodology

- for stochastic line balancing with intermittent line stoppages", *Man.Sci.*, vol.32, no.4, 1986.
- Sniedovich, M., " Analysis of preference order assembly line problem", *Man.Sci.*, vol.27, no.9, 1981.
- Sphicas, G.P., Silverman, F.N., " Deterministic equivalents for stochastic assembly line balancing", *AIIE Transactions*, vol.8, no.2, 1976.
- Sumichrast, R.T., Russel, R.A., Taylor, B.W., " A comparative analysis of sequencing procedures for mixed-model assembly lines in a Just-In-Time production system", *IJPR*, vol.30, no.1, 1992.
- Talbot, F.B., Patterson, J.F., " An integer programming algorithm with network cuts for solving the assembly line balancing problem", *Man.Sci.*, vol.30, no.1, 1984.
- Thangavelu, S.R., Shetty, C.M., " Assembly line balancing by zero-one programming", *AIIE Transactions*, vol.3, no.1, 1971.
- Thomopoulos, N.T., " Line balancing-sequencing for mixed model assembly", *Man.Sci.*, vol.14, no.2, 1967.
- Thomopoulos, N.T., " Mixed-model line balancing with smoothed station assignments", *Man.Sci.*, vol.16, no.9, 1970.
- Tonge, F.M., " Summary of a heuristic line balancing procedure", *Man.Sci.*, vol.7, no.1, 1960.
- Ulusoy, G., Büyükbacı, M., Bekar, Z., " Montaj hattı

- dengelemesinde bir uygulama", 2.Ulusal Makina
Tasarım ve İmalat Kongresi, ODTÜ, Ankara, 1986.
- Van Assche, F., Herroelen, W.S., " An optimal procedure for
the single-model deterministic assembly line
balancing problem", European Jour. of Operation
research, vol.3, no.2, 1979.
- Vrat, P., Virani, A., " A cost model for optimal mix of
balanced stochastic assembly line and the modular
assembly system for a customer oriented production
system", IJPR, vol.14, no.4, 1976.
- Wee, T.S., Magazine, M.J., " An efficient branch and bound
algorithm for an assembly line balancing problem.
Part I: minimize the number of work stations",
Working paper, no:150, University of Waterloo,
Ontario, Canada, June, 1981.
- Wee, T.S., Magazine, M.J., " Assembly line balancing as
generalized bin packing", OR Letters, vol.1, no.2,
1982.
- Wild, R., " Mass-Production Management", J.Wiley & Sons
Ltd, New York, 1972.
- Wild, R., " Production & Operations Management", fourth
edition, Great Britain, 1989.

EK-1

Birleřtirilmiř Diyagramda, Herbir Grev iin E_i ve L_i Sınırlarını Bulan *Program EİLi* 'nin FORTRAN Diliyle Kodlanmış Yazılımı ve rnek Bir Problem iin zm ıktısı.

Bu program, 0-1 modellemesi için gerekli olan ve Program WKM ye girdi olarak istenen E_i ve L_i hesaplamalarını yapar. Program, dataların kütükten okunmasına imkan verir. Geliştirilen sezgisel metottan (Program MıxalB) elde edilen istasyon sayısı (K veya KST) 'nin Program EİLi'ye girdi olarak verilmesi gerekmektedir.

Program EİLi

```
DIMENSION NP(65,10),C(210),D(60,900),TIME(65,10)
DIMENSION IPS(65),ARA(65),FARK(65),F(65),AR(65)
DIMENSION PG(65),CYC(4),FAR(65),FA(65),E(90),B(210)
DIMENSION BB(200),CC(200),DD(60,500),EE(90),IPW(65)
INTEGER BB,CC,Z,DD,EE,IPW
INTEGER NP,B,C,S,T,L1,D,E,IPS,TIME,TOPI,TOPI,SUM,CYC
REAL EI(65),LI(65),CTOP
OPEN(1,FILE='DENE.DAT')
OPEN(2,FILE='DENE.REP')
READ(1,*) NOP,IS,MXP,(CYC(IU)),IU=1,NOP)
WRITE(*,*) NOP,IS,MXP,(CYC(IU)),IU=1,NOP)
DO 1402 I3=1,IS
1402 READ(1,*) (TIME(I3,IU)),IU=1,NOP)
DO 1403 J=1,IS
1403 READ(1,*) (NP(J,K),K=1,MXP)
CTOP=0
SUM=0
S=0
DO 76 IC=1,NOP
76 CTOP=CTOP+CYC(IC)
DO 3 I=1,IS
L1=0
DO 4 K=1,MXP
IF (NP(I,K).NE.0) GOTO 5
GOTO 4
5 S=S+1
L1=L1+1
```

```

      D(I,L1)=NP(I,K)
      B(S)=NP(I,K)
4    CONTINUE
      IF(S.NE.0) GOTO 20
      NP(I,K)=0
      WRITE(*,40) I
40   FORMAT(I2,2X,'.PS YE AIT ELEMENLAR')
      WRITE(*,*)'-----'
      WRITE(*,*) NP(I,K)
      WRITE(*,*) I
      DO 77 IU=1,NOP
77   SUM=SUM+TIME(I,IU)
      IPS(I)=SUM
      EI(1)=1
      WRITE(*,78) I,IPS(I),EI(I),CTOP
78   FORMAT(I2,'.PS=',2X,I3,2X,'EI(I)=',F5.2,F5.2)
      S=0
      GOTO 3
20   CALL PS(NP,B,S,D,L1,I,MXP)
      IF(S.NE.0) GOTO 20
      S=0
      E(I)=L1
3    CONTINUE
      TOP1=0
      TOPI=0
      DO 31 I=2,IS
      DO 32 J=1,E(I)
      KK=J+1
      IF(KK.GT.E(I)) GOTO 32
35   IF(D(I,J).EQ.D(I,KK)) GOTO 33
      GOTO 34
33   D(I,KK)=0
34   KK=KK+1
      IF(KK.GT.E(I)) GOTO 32
      GOTO 35
32   CONTINUE

```



```

31 CONTINUE
    DO 21 I=2,IS
        WRITE(*,41)I
41  FORMAT(I2,2X,'PS YE AIT ELEMENLAR:')
        WRITE(*,*)'-----'
        WRITE(*,*) I
        DO 22 J=1,E(I)
            IF(D(I,J).EQ.0) GOTO 22
        DO 51 IU=1,NOP
51  TOP1=TOP1+TIME(D(I,J),IU)
        WRITE(*,*) D(I,J)
22  CONTINUE
        DO 52 IU=1,NOP
52  TOPI=TOPI+TIME(I,IU)
        IPS(I)=TOP1+TOPI
        EI(I)=IPS(I)/CTOP
        ARA(I)=INT(EI(I))
        IF(EI(I).EQ.ARA(I)) GOTO 79
        FARK(I)=EI(I)-ARA(I)
        IF(FARK(I).GT.0) GOTO 80
        GOTO 21
80  F(I)=ARA(I)+1
        EI(I)=F(I)
79  WRITE(*,10)IPS(I),EI(I),CTOP
10  FORMAT('PS=',2X,I3,2X,'EI(I)=' ,F5.2,2X,'CTOP=' ,F5.2)
        WRITE(*,*) '**EI=[PS/CTOP]+ **'
        TOP1=0
        TOPI=0
21  CONTINUE
    S=0
C    KST, Program Mıxal b 'den elde edilen istasyon sayısı
    KST=4
    Z=1
    L1=0
555 DO 3333 I=2,IS
    DO 44 K=1,MXP

```

```

        IF (NP(I,K).EQ.Z) GOTO 55
        GOTO 44
55 S=S+1
    L1=L1+1
    DD(Z,L1)=I
    BB(S)=I
44 CONTINUE
3333 CONTINUE
220 CALL PW(NP,BB,S,DD,Z,L1,IS,MXP)
    IF(S.NE.0) GOTO 220
    S=0
    EE(Z)=L1
    IF (Z.EQ.IS) GOTO 1100
    Z=Z+1
    L1=0
    GOTO 555
1100 TOP1=0
    TOPI=0
    DO 331 Z=1,IS
        DO 332 J=1,EE(Z)
            KK=J+1
            IF (KK.GT.EE(Z)) GOTO 332
335 IF(DD(Z,J).EQ.DD(Z,KK)) GOTO 333
            GOTO 334
333 DD(Z,KK)=0
334 KK=KK+1
            IF (KK.GT.EE(Z)) GOTO 332
            GOTO 335
332 CONTINUE
331 CONTINUE
    DO 221 Z=1,IS
        WRITE(*,441)Z
441 FORMAT(I2,2X,'PW YE AIT ELEMENLAR:')
        WRITE(*,*)'-----'
        WRITE(*,*) Z
    DO 222 J=1,EE(Z)

```

```

        IF(DD(Z,J).EQ.0) GOTO 222
        DO 551 IU=1,NOP
551    TOP1=TOP1+TIME(DD(Z,J),IU)
        WRITE(*,*) DD(Z,J)
222    CONTINUE
        DO 552 IU=1,NOP
552    TOPI=TOPI+TIME(Z,IU)
        IPW(Z)=TOP1+TOPI
        PG(Z)=IPW(Z)/CTOP
        AR(Z)=INT(PG(Z))
        IF(PG(Z).EQ.AR(Z)) GOTO 72
        FAR(Z)=PG(Z)-AR(Z)
        IF(FAR(Z).GT.0) GOTO 73
        GOTO 221
73    FA(Z)=AR(Z)+1
        PG(Z)=FA(Z)
72    LI(Z)=KST+1-PG(Z)
74    WRITE(*,111)Z,IPW(Z),CTOP,LI(Z),KST
111    FORMAT(I2,'.PW=',I3,2X,'CTOP=',F5.2,2X,
        *' LI(Z)=' ,F5.2,2X,'KST=',I2)
        WRITE(*,*) '**LI=KST+1-[PW/CTOP]+ **'
        TOP1=0
        TOPI=0
221    CONTINUE
        WRITE(*,*) '*****Özet Rapor*****'
        WRITE(2,*) '***** Özet Rapor *****'
        DO 28 I=1,IS
            WRITE(*,29) I,INT(EI(I)),I,INT(LI(I))
29    FORMAT(2X,'Ei(',I2,')=' ,I2,'====',1X,
        * 'Li(',I2,')=' ,I2)
94    WRITE(2,95) I,INT(EI(I)),I,INT(LI(I))
95    FORMAT(2X,'Ei(',I2,')=' ,I2,'====^P',1X,
        * 'Li(',I2,')=' ,I2)
28    CONTINUE
        END

```

```

SUBROUTINE PW (NP,BB,S,DD,Z,L1,IS,MXP)
DIMENSION NP(65,10),BB(200),CC(200),DD(60,500),EE(50)
INTEGER NP,BB,CC,S,T,Z,DD,L1
T=0
DO 66 J=1,S
DO 66 I=1,IS
DO 66 K=1,MXP
IF(NP(I,K).EQ.BB(J)) GOTO 88
GOTO 66
88 T=T+1
CC(T)=I
66 CONTINUE
DO 99 L=1,T
L1=L1+1
DD(Z,L1)=CC(L)
99 BB(L)=CC(L)
S=T
RETURN
END

```

```

SUBROUTINE PS(NP,B,S,D,L1,I,MXP)
DIMENSION NP(65,10),B(210),C(210),D(60,900),E(90)
INTEGER NP,B,C,S,T,D,L1
T=0
DO 6 J=1,S
DO 6 K=1,MXP
IF(NP(B(J),K).NE.0) GOTO 8
GOTO 6
8 T=T+1
C(T)=NP(B(J),K)
6 CONTINUE
IF(T.NE.0) GOTO 92
S=0
RETURN
92 DO 9 L=1,T
L1=L1+1

```

```
D(I,L1)=C(L)  
9 B(L)=C(L)  
S=T  
RETURN  
END
```



4 Ürünlü, çevrim zamanı her bir ürün için 12, sezgiselden (Program Mıxalib) elde edilen K (veya KST) değerinin 8 olduğu 30 görevli bir problem için, Program Eili den elde edilen En Erken (E_i) ve En Geç istasyon (L_i) değerleri aşağıdaki gibidir.

*****ÖZET RAPOR*****

Ei(1)= 1 =====> Li(1)= 3
Ei(2)= 1 =====> Li(2)= 6
Ei(3)= 1 =====> Li(3)= 7
Ei(4)= 1 =====> Li(4)= 7
Ei(5)= 1 =====> Li(5)= 7
Ei(6)= 1 =====> Li(6)= 7
Ei(7)= 2 =====> Li(7)= 7
Ei(8)= 1 =====> Li(8)= 7
Ei(9)= 1 =====> Li(9)= 8
Ei(10)= 1 =====> Li(10)= 8
Ei(11)= 2 =====> Li(11)= 8
Ei(12)= 2 =====> Li(12)= 8
Ei(13)= 2 =====> Li(13)= 8
Ei(14)= 1 =====> Li(14)= 8
Ei(15)= 2 =====> Li(15)= 8
Ei(16)= 2 =====> Li(16)= 7
Ei(17)= 1 =====> Li(17)= 8
Ei(18)= 2 =====> Li(18)= 8
Ei(19)= 1 =====> Li(19)= 8
Ei(20)= 1 =====> Li(20)= 8
Ei(21)= 2 =====> Li(21)= 8
Ei(22)= 2 =====> Li(22)= 8
Ei(23)= 1 =====> Li(23)= 8
Ei(24)= 2 =====> Li(24)= 8
Ei(25)= 2 =====> Li(25)= 8
Ei(26)= 2 =====> Li(26)= 8
Ei(27)= 2 =====> Li(27)= 8
Ei(28)= 3 =====> Li(28)= 8
Ei(29)= 3 =====> Li(29)= 8
Ei(30)= 6 =====> Li(30)= 8

EK-2

**Bir Karışık Modelli MHD Probleminin 0-1 Saf (Pure)
Tamsayı Programlama Modeli ve Çözüm Çıktısı**

Model 2.1

Amaç Fonksiyonu

$$\min A1+A2+A3+A4$$

S.t

Atama Kısıtları

$$V11+V12=1$$

$$V21+V22+V23+V24=1$$

$$V31+V32+V33+V34=1$$

$$V41+V42+V43=1$$

$$V51+V52+V53=1$$

$$V61+V62+V63+V64=1$$

$$V72+V73+V74=1$$

$$V81+V82+V83=1$$

$$V91+V92+V93+V94=1$$

$$V101+V102+V103+V104=1$$

$$V113+V114=1$$

Öncelik Kısıtları

$$V11+2V12-V31-2V32-3V33-4V34\leq 0$$

$$V11+2V12-V41-2V42-3V43\leq 0$$

$$V11+2V12-V21-2V22-3V23-4V24\leq 0$$

$$V11+2V12-V81-2V82-3V83\leq 0$$

$$V41+2V42+3V43-V51-2V52-3V53\leq 0$$

$$V51+2V52+3V53-V61-2V62-3V63-4V64\leq 0$$

$$V31+2V32+3V33+4V34-2V72-3V73-4V74\leq 0$$

$$V61+2V62+3V63+4V64-2V72-3V73-4V74\leq 0$$

$$V21+2V22+3V23+4V24-2V72-3V73-4V74\leq 0$$

$$V81+2V82+3V83-V91-2V92-3V93-4V94\leq 0$$

$$V91+2V92+3V93+4V94-V101-2V102-3V103-4V104\leq 0$$

$$2V72+3V73+4V74-3V113-4V114\leq 0$$

$$V101+2V102+3V103+4V104-3V113-4V114\leq 0$$

Çevrim Zamanı Kısıtları (1. ürün için)

$$V11+5V21+4V31+4V81+3V91\leq 10$$

$$V12+5V22+4V32+2V72+4V82+3V92\leq 10$$

$$5V23+4V33+2V73+4V83+3V93+3V113\leq 10$$

$$5V24+4V34+2V74+3V94+3V114\leq 10$$

Çevrim Zamanı Kısıtları (2. ürün için)

$$V11+4V31+V41+5V51+6V61+3V91+5V101 \leq 10$$

$$V12+4V32+V42+5V52+6V62+2V72+3V92+5V102 \leq 10$$

$$4V33+V43+5V53+6V63+2V73+3V93+5V103+3V113 \leq 10$$

$$4V34+6V64+2V74+3V94+5V104+3V114 \leq 10$$

İş İstasyon Kısıtları

$$V11+V21+V31+V81+V91-5X11 \leq 0$$

$$V12+V22+V32+V72+V82+V92-6X21 \leq 0$$

$$V23+V33+V73+V83+V93+V113-6X31 \leq 0$$

$$V24+V34+V74+V94+V114-5X41 \leq 0$$

$$V11+V31+V41+V51+V61+V91+V101-7X12 \leq 0$$

$$V12+V32+V42+V52+V62+V72+V92+V102-8X22 \leq 0$$

$$V33+V43+V53+V63+V73+V93+V103+V113-8X32 \leq 0$$

$$V34+V64+V74+V94+V104+V114-6X42 \leq 0$$

$$X11+X12-2A1=0$$

$$X21+X22-2A2=0$$

$$X31+X32-2A3=0$$

$$X41+X42-2A4=0$$

End

Tüm X_{km} 0 veya 1

Tüm V_{ik} 0 veya 1

Tüm A_k 0 veya 1

Model 2.1 'in çalıştırılması sonucu elde edilen ikili değişkenlerin ve amaç fonksiyonunun değeri aşağıda verilmiştir:

OBJECTIVE FUNCTION VALUE

1) 3.00000000

VARIABLE	VALUE	REDUCED COST
A1	1.000000	1.000000
A2	.000000	1.000000
A3	1.000000	1.000000
A4	1.000000	1.000000
V11	1.000000	.000000
V12	.000000	.000000
V21	.000000	.000000
V22	.000000	.000000
V23	.000000	.000000
V24	1.000000	.000000
V31	.000000	.000000
V32	.000000	.000000
V33	1.000000	.000000
V34	.000000	.000000
V41	1.000000	.000000
V42	.000000	.000000
V43	.000000	.000000
V51	1.000000	.000000
V52	.000000	.000000
V53	.000000	.000000
V61	.000000	.000000
V62	.000000	.000000
V63	1.000000	.000000
V64	.000000	.000000
V72	.000000	.000000
V73	.000000	.000000
V74	1.000000	.000000

V81	1.000000	.000000
V82	.000000	.000000
V83	.000000	.000000
V91	1.000000	.000000
V92	.000000	.000000
V93	.000000	.000000
V94	.000000	.000000
V101	.000000	.000000
V102	.000000	.000000
V103	.000000	.000000
V104	1.000000	.000000
V113	.000000	.000000
V114	1.000000	.000000
X11	1.000000	.000000
X21	.000000	.000000
X31	1.000000	.000000
X41	1.000000	.000000
X12	1.000000	.000000
X22	.000000	.000000
X32	1.000000	.000000
X42	1.000000	.000000

EK-3

Geliştirilen Sezgisel Yöntemin (*Program MixA1b*) FORTRAN
Diliyle Kodlanmış Yazılımı ve Örnek Bir Problem İçin
Çözüm Çıktısı

Bu program, tek ürünler için geliştirilen RPW sezgiseli baz alınarak geliştirilmiştir. Program, kütükten çalışmaya ya da bilgisayarla etkileşimli olarak çalışılmasına izin verir. Program, her bir görevin istasyon atamalarından, ürün bazında ve genel olarak hattın etkinliği ve gecikmelerinden oluşan detaylı bir rapor sunar. Programdaki Dimensionların boyutu, problem boyutuna göre ayarlanabilir.

Program MixAlb

```
DIMENSION NP(80,9),IADAY(70,80),KADAY(70,80)
DIMENSION ICIKAN(70,80), IPW(62)
DIMENSION KALAN(10),ADAY(10),CYC(10),ISEC(30)
DIMENSION TOPB(5),TOPA(5),GEN(5),BOS(5),VER(5)
DIMENSION B(4000),D(60,200),E(80),C(300)
INTEGER B,S,T,ZZ,LH1,D,E,TOPI,HS,C
INTEGER IONC(30),DIZ(21),IKADAY(70,80),DDIZ(30),
INTEGER IMAX(70), IG(30), IDIZ(21), PW(62)
INTEGER TIME(62,5),YD(70,80),DZ(30,30)
OPEN(1,FILE='URUN32.DAT'):OPEN(2,FILE='URUN32.REP')
WRITE(*,*)'          M E N U          '
WRITE(*,*)' [ ] '
WRITE(*,*)'| 1.Yeni Veri Girisi(kutuge Kayit) |'
WRITE(*,*)'| 2.Kutukteki Veriyle Calisma |'
WRITE(*,*)'| 3.Cikis |'
WRITE(*,*)'| [ ] '
1405 WRITE(*,*)'SECIMINIZ : '
READ(*,*) ICH
IF(ICH.EQ.1) GOTO 1406
IF(ICH.EQ.2) GOTO 1407
IF(ICH.EQ.3) GOTO 803
GOTO 1405
1406 WRITE(*,1)
1 FORMAT('URUN SAYISI NEDIR-->')
READ (*,*) NOP
DO 51 IU=1,NOP
```

```

        WRITE (*,2) IU
2  FORMAT(2X,I2,'. URUNUN CYC DEGERI-->')
51 READ (*,*) CYC(IU)
    WRITE (*,3)
3  FORMAT('TOPLAM IS EL.SAYISI=')
    READ (*,*) IS
    DO 53 MI=1,IS
    DO 53 IU=1,NOP
11 WRITE(*,4) MI,IU
    4  FORMAT(2X,I2,' ELEMANIN ZAMANI (',I2,'.URUN ICIN)')
53 READ(*,*) TIME(MI,IU)
    WRITE(*,330)
330 FORMAT(2X,'MAX ONCEL SAYISI-->')
    READ(*,*) MXP
    DO 100 J=1,IS
    WRITE(*,350) J
350 FORMAT(2X,I3,'.ELEMANIN ONCULLERINI GIR-->')
    READ(*,*) (NP(J,K),K=1,MXP)
100 CONTINUE
    WRITE(*,*)' kutuge yaziliyor'
    WRITE(1,*) NOP,IS,MXP,(CYC(IU),IU=1,NOP)
    DO 1400 MI=1,IS
1400 WRITE(1,*) (TIME(MI,IU),IU=1,NOP)
    DO 1401 J=1,IS
1401 WRITE(1,*) (NP(J,K),K=1,MXP)
    CLOSE(1)
1407 OPEN(1,FILE='URUN32.DAT')
    READ(1,*) NOP,IS,MXP,(CYC(IU),IU=1,NOP)
    DO 1402 MI=1,IS
1402 READ(1,*) (TIME(MI,IU),IU=1,NOP)
    DO 1403 J=1,IS
1403 READ(1,*) (NP(J,K),K=1,MXP)
    DO 52 IU=1,NOP
    KALAN(IU)=CYC(IU)
    TOPA(IU)=0
52 TOPB(IU)=0

```

```
KKK=1
MM1=1
KALTOP=0
KALSOP=0
CTOP=0
SUM=0
SU=0
L=1
IC=1
IT=1
DO 800 I=1,IS
DO 800 IU=1,NOP
IF(TIME(I,IU).GT.CYC(IU)) GOTO 801
800 CONTINUE
GOTO 804
801 WRITE(*,*)'Zaman çevrim zamanı geçiyor'
WRITE(*,*)'Lütfen zamanları kontrol et !!!'
GOTO 803
804 S=0
ZZ=1
LH1=0
7055 DO 703 I=1,IS
DO 704 K=1,MXP
IF (NP(I,K).EQ.ZZ) GOTO 705
IF (NP(I,K).EQ.0) GOTO 703
GOTO 704
705 S=S+1
LH1=LH1+1
D(ZZ,LH1)=I
B(S)=I
704 CONTINUE
703 CONTINUE
7020 CALL PWT(NP,B,S,D,ZZ,LH1,IS,MXP)
IF(S.NE.0) GOTO 7020
E(ZZ)=LH1
IF (ZZ.EQ.IS) GOTO 7100
```

```

        ZZ=ZZ+1
        LH1=0
        GOTO 7055
7100  TOP1=0
        TOPI=0
        DO 7031 I=1,IS
        DO 7032 J=1,E(I)
        NN=J+1
        IF (NN.GT.E(I)) GOTO 7032
7035  IF(D(I,J).EQ.D(I,NN)) GOTO 7033
        GOTO 7034
7033  D(I,NN)=0
7034  NN=NN+1
        IF (NN.GT.E(I)) GOTO 7032
        GOTO 7035
7032  CONTINUE
7031  CONTINUE
        DO 7021 I=1,IS
        DO 7022 J=1,E(I)
        IF(D(I,J).EQ.0) GOTO 7022
        DO 7051 IU=1,NOP
7051  TOP1=TOP1+TIME(D(I,J),IU)
7022  CONTINUE
        DO 7052 IU=1,NOP
7052  TOPI=TOPI+TIME(I,IU)
        PW(I)=TOP1+TOPI
        TOP1=0
        TOPI=0
7021  CONTINUE
        DO 101 J=1,IS
        DO 101 K=1,MXP
101  IF (NP(J,K).EQ.0) GOTO 105
        GOTO 108
105  GOTO 111
108  WRITE(*,5)
        5 FORMAT('Başlangıç düğümü yoktur')

```



```

        GOTO 1000
111  IONC(L)=J
      IADAY(IT,L)=IONC(L)
      MAX=1
      WRITE(*,*) ' _____ '
      WRITE(2,*) ' _____ '
      WRITE(*,300) MAX,KKK
300  FORMAT('||',1X,I2,'. ELEMEN',2X,I2,
* '.ISTASYONA ATANIR',2X,'||')
      WRITE(2,302) MAX,KKK
302  FORMAT('||',1X,I2,'. ELEMEN',2X,I2,
* '.ISTASYONA ATANIR',2X,'||')
      DO 310 IU=1,NOP
310  KALAN(IU)=KALAN(IU)-TIME(MAX,IU)
      YD(KKK,MM1)=MAX
      IT=IT+1
      L1=0
      DO 121 J=1,IS
      DO 121 K=1,MXP
      IF(NP(J,K).EQ.MAX) GOTO 115
      GOTO 121
115  L=L+1
      L1=L1+1
      DIZ(L1)=L
      IONC(L)=J
      IADAY(IT,L)=IONC(L)
121  CONTINUE
      N2=L
      IA=DIZ(1)
      NN2=N2
      II=1
      IMAX(1)=1
      GOTO 666
999  DO 30 I=1,IP-1
      LIL=I+1
      DO 30 KIL=LIL,IP

```

```

      IF(IPW(KIL).LE.IPW(I)) GOTO 30
      HS=IPW(I)
      D2=ISEC(I)
      IPW(I)=IPW(KIL)
      ISEC(I)=ISEC(KIL)
      IPW(KIL)=HS
      ISEC(KIL)=D2
30  CONTINUE
56  I=1
577 MAX=ISEC(I)
      IU=1
20  IF (TIME(MAX,IU).LE.KALAN(IU)) GOTO 54
      GOTO 15
54  IU=IU+1
      IF (IU.LE.NOP) GOTO 20
      9  WRITE(*,10) MAX,KKK
10  FORMAT('||',1X,I2,'.  ELEMAN',2X,I2,
*  '.ISTASYONA ATANIR',2X,'||')
      WRITE(2,107) MAX,KKK
107 FORMAT('||',1X,I2,'.  ELEMAN',2X,I2,
*  '.ISTASYONA ATANIR',2X,'||')
      LLL=KKK
      MM1=MM1+1
      YD(LLI,MM1)=MAX
      DO 55 IU=1,NOP
55  KALAN(IU)=KALAN(IU)-TIME(MAX,IU)
      GOTO 340
15  I=I+1
      IF(I.GT.IP) GOTO 18
      MAX=ISEC(I)
      GOTO 577
18  DO 168 IU=1,NOP
      SUM=SUM+KALAN(IU)
      TOPB(IU)=TOPB(IU)+SUM
      SUM=0
168  KALTOP=KALTOP+KALAN(IU)

```

```

      KKK=KKK+1
      DO 57 IU=1,NOP
57    KALAN(IU)=CYC(IU)
      GOTO 666
340   IF(MAX.EQ.IS) GOTO 1000
      IF(MAX.EQ.0) GOTO 1000
      II=II+1
      IMAX(II)=MAX
      IM=II
      IT=IT+1
      LL1=0
      DO 128 J=1,IS
      DO 128 K=1,MXP
      IF(NP(J,K).EQ.MAX) GOTO 116
      GOTO 128
116   L=L+1
      LL1=LL1+1
      IDIZ(LL1)=L
      IG(LL1)=J
128   CONTINUE
      K1=IDIZ(1)
      K2=L
      DO 776 NA=1,LL1
      IADAY(IT,K1)=IG(NA)
      KK=0
      LF=0
      DO 200 IZ=1,MXP
      DO 201 IR=2,IM
      IF(NP(IG(NA),IZ).NE.0) GO TO 202
      GO TO 200
202   IF(NP(IG(NA),IZ).EQ.IMAX(IR)) GO TO 203
      IF(IR.NE.IM) GO TO 201
      IADAY(IT,K1)=0
      GO TO 185
203   KK=IZ+1
      LF=LF+1

```

```

        IF(KK.GT.MXP) GO TO 200
        IF(NP(IG(NA),KK).NE.0) GO TO 200
        IADAY(IT,K1)=IG(NA)
        GO TO 185
201  CONTINUE
200  CONTINUE
        IF(LF.EQ.MXP) GO TO 250
        IADAY(IT,K1)=0
        GO TO 185
250  IADAY(IT,K1)=IG(NA)
185  K1=K1+1
        IF(K1.GT.K2) GOTO 696
776  CONTINUE
696  L=K1-1
        NN2=L
        IT=IT-1
        DO 129 I2=IA,N2
        IIT=IT+1
        IF (IADAY(IT,I2).EQ.MAX) GOTO 117
        KADAY(IIT,I2)=IADAY(IT,I2)
        IADAY(IIT,I2)=KADAY(IIT,I2)
        GOTO 129
117  ICIKAN(IIT,I2)=IADAY(IT,I2)
        IADAY(IIT,I2)=ICIKAN(IIT,I2)
        IADAY(IIT,I2)=0
129  CONTINUE
        IT=IIT
        N2=NN2
666  IP=0
        DO 130 I3=IA,NN2
        IF(IADAY(IT,I3).EQ.0) GOTO 130
        IP=IP+1
        IPW(IP)=PW(IADAY(IT,I3))
        ISEC(IP)=IADAY(IT,I3)
130  CONTINUE
        IF(IP.EQ.0) GOTO 1000

```

```

      GOTO 999
1000 WRITE(*,*) ' _____'
      WRITE(2,*) ' _____'
      WRITE(*,34) LLL
34   FORMAT(3X,'ISTASYON SAYISI----->',I2)
      WRITE(2,349) LLL
349  FORMAT(3X,'ISTASYON SAYISI----->',I2)
      DO 787 IU=1,NOP
      KALSOP=KALSOP+KALAN(IU)
      SU=SU+KALAN(IU)
      TOPA(IU)=TOPA(IU)+SU
787  SU=0
      DO 789 IU=1,NOP
789  GEN(IU)=TOPB(IU)+TOPA(IU)
      DO 47 IU=1,NOP
      BOS(IU)=(GEN(IU)/(LLL*CYC(IU)))*100
      VER(IU)=100-BOS(IU)
      WRITE(*,675)IU,BOS(IU)
675  FORMAT('||',I2,'. URUN ICIN HATTIN GECIKMESI
*  %',F7.3,2X,'||')
      WRITE(2,685)IU,BOS(IU)
685  FORMAT('||',I2,'. URUN ICIN HATTIN GECIKMESI
*  %',F7.3,2X,'||')
      WRITE(*,676)IU,VER(IU)
      WRITE(2,686)IU,VER(IU)
      WRITE(*,*)' || _____ ||'
676  FORMAT('||',I2,'. URUN ICIN HATTIN ETKINLIGI
*  %',F7.3,2X,'||')
      WRITE(2,*)' || _____ ||'
686  FORMAT('||',I2,'. URUN ICIN HATTIN ETKINLIGI
*  %',F7.3,2X,'||')
47   CONTINUE
      NETTOP=KALTOP+KALSOP
      DO 444 IU=1,NOP
444  CTOP=CTOP+CYC(IU)
      DELAY=(NETTOP/(LLL*CTOP))*100

```

```

      UTIL=100-DELAY
      WRITE(*,445) DELAY
445  FORMAT('||',' HATTIN GENEL TOPLAM GECIKMESI
      *  %',F7.3,2X,'||')
      WRITE(2,448) DELAY
448  FORMAT('||',' HATTIN GENEL TOPLAM GECIKMESI
      *  %',F7.3,2X,'||')
      WRITE(*,446) UTIL
446  FORMAT('||',' HATTIN GENEL TOPLAM ETKINLIGI
      *  %',F7.3,2X,'||')
      WRITE(2,449) UTIL
449  FORMAT('||',' HATTIN GENEL TOPLAM ETKINLIGI
      *  %',F7.3,2X,'||')
803  END

```

```

      SUBROUTINE PWT (NP,B,S,D,ZZ,LH1,IS,MXP)
      DIMENSION NP(80,9),B(4000),E(80),D(60,200),C(300)
      INTEGER NP,B,S,T,ZZ,D,LH1,C
      IF(S.LE.1) GOTO 22
      GOTO 23
22  T=0
      DO 5 J=1,S
      DO 6 I=1,IS
      DO 7 K=1,MXP
      IF(NP(I,K).EQ.B(J)) GOTO 8
      IF(NP(I,K).EQ.0) GOTO 6
      GOTO 7
8  T=T+1
      C(T)=I
7  CONTINUE
6  CONTINUE
5  CONTINUE
      DO 9 JL=1,T
      LH1=LH1+1
      D(ZZ,LH1)=C(JL)
9  B(JL)=C(JL)

```

```

      S=T
      RETURN
23  T=0
      DO 51 J=1,S
      DO 61 I=1,IS
      DO 71 K=1,MXP
      IF((B(J).NE.0).AND.(NP(I,K).EQ.B(J))) GOTO 81
      IF(NP(I,K).EQ.0) GOTO 61
      GOTO 71
81  T=T+1
      C(T)=I
71  CONTINUE
61  CONTINUE
51  CONTINUE
      DO 91 I1=1,T
      NIN=I1+1
      IF(NIN.GT.T) GOTO 91
101 IF(C(I1).EQ.C(NIN)) GOTO 111
      GOTO 121
111 C(NIN)=0
121 NIN=NIN+1
      IF(NIN.GT.T) GOTO 91
      GOTO 101
91  CONTINUE
      IR=0
      DO 131 L=1,T
      IF(C(L).EQ.0) GOTO 131
      IR=IR+1
      LH1=LH1+1
      D(ZZ,LH1)=C(L)
      C(IR)=D(ZZ,LH1)
      B(IR)=C(IR)
131 CONTINUE
      S=IR
      RETURN
      END

```

4 ürünlü, 30 görevli (Birleştirilmiş Diyagramda) bir problem için Program Mixalb 'in çalıştırılması sonucu elde edilen neticeler şöyledir.

1. ELEMAN	1. ISTASYONA ATANIR
2. ELEMAN	1. ISTASYONA ATANIR
4. ELEMAN	1. ISTASYONA ATANIR
3. ELEMAN	1. ISTASYONA ATANIR
8. ELEMAN	2. ISTASYONA ATANIR
7. ELEMAN	2. ISTASYONA ATANIR
6. ELEMAN	2. ISTASYONA ATANIR
10. ELEMAN	2. ISTASYONA ATANIR
18. ELEMAN	2. ISTASYONA ATANIR
19. ELEMAN	2. ISTASYONA ATANIR
16. ELEMAN	3. ISTASYONA ATANIR
5. ELEMAN	3. ISTASYONA ATANIR
11. ELEMAN	3. ISTASYONA ATANIR
14. ELEMAN	3. ISTASYONA ATANIR
17. ELEMAN	4. ISTASYONA ATANIR
23. ELEMAN	4. ISTASYONA ATANIR
21. ELEMAN	4. ISTASYONA ATANIR
22. ELEMAN	4. ISTASYONA ATANIR
24. ELEMAN	5. ISTASYONA ATANIR
27. ELEMAN	5. ISTASYONA ATANIR
9. ELEMAN	5. ISTASYONA ATANIR
26. ELEMAN	6. ISTASYONA ATANIR
28. ELEMAN	6. ISTASYONA ATANIR
12. ELEMAN	6. ISTASYONA ATANIR
15. ELEMAN	6. ISTASYONA ATANIR
13. ELEMAN	7. ISTASYONA ATANIR
20. ELEMAN	7. ISTASYONA ATANIR
29. ELEMAN	7. ISTASYONA ATANIR
25. ELEMAN	7. ISTASYONA ATANIR
30. ELEMAN	8. ISTASYONA ATANIR

ISTASYON SAYISI-----> 8

1. URUN ICIN HATTIN GECIKMESI	% 22.917
1. URUN ICIN HATTIN ETKINLIGI	% 77.083
2. URUN ICIN HATTIN GECIKMESI	% 16.667
2. URUN ICIN HATTIN ETKINLIGI	% 83.333
3. URUN ICIN HATTIN GECIKMESI	% 35.417
3. URUN ICIN HATTIN ETKINLIGI	% 64.583
4. URUN ICIN HATTIN GECIKMESI	% 25.000
4. URUN ICIN HATTIN ETKINLIGI	% 75.000
HATTIN GENEL TOPLAM GECIKMESI	% 25.000
HATTIN GENEL TOPLAM ETKINLIGI	% 75.000

EK-4
İki Ürünlü Karışık MHD Problemi İçin Oluşturulan
GAMS Modeli

İki ürünli ve birleştirilmiş diyagramda 11 görevi olan karışık MHD problemlerinin, 0-1 tamsayılı programlama şeklinde oluşturulması için geliştirilen GAMS modeli:

I : Birleştirilmiş diyagramdaki düğüm(görev) sayısı.

K : Müsade edilen iş istasyon sayısı

SAYIA(K) : 1. ürün için, k istasyonuna atanmaya aday düğüm(görev) lerin sayısı.

SAYIB(K) : 2. ürün için, k istasyonuna atanmaya aday düğüm(görev) lerin sayısı

KAT(K) : Öncelik kısıtında, her hangi iki düğüm arasındaki ilişkinin ifade edilmesinde kullanılacak katsayıları ifade eder.

TABLE ASS(I,K) : Atama kısıtındaki katsayıları belirleyerek kısıt oluşumunu temin eder. Ayrıca öncelik kısıtlarının oluşturulması da düzenler.

TABLE TIME1(K,I) : 1. ürün için, çevrim zaman kısıt katsayılarıdır. (K=1,...,4)

TABLE TIME3(K,I) : 2. ürün için çevrim zaman kısıt katsayılarıdır. (K=1,...,4)

TABLE TIME2(K,I) : 1. ürün için iş istasyon kısıt katsayılarıdır. (K=1,...,4)

TABLE TIME4(K,I) : 2. ürün için iş istasyon kısıt katsayılarıdır. (K=1,...,4)

GAMS Model

SETS

I /1,2,3,4,5,6,7,8,9,10,11/

K /1,2,3,4/;

PARAMETERS SAYIA(K)

/1 5

2 6

3 6

4 5/

SAYIB(K)

/1 7

2 8

3 8

4 6/

KAT(K)

/1 1

2 2

3 3

4 4/;

TABLE ASS(I,K)

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

TABLE TIME1(I,K)

	1	2	3	4
1	1	1	0	0
2	5	5	5	5
3	4	4	4	4
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	2	2	2
8	4	4	4	0
9	3	3	3	3
10	0	0	0	0
11	0	0	3	3

TABLE TIME3(K,I)

	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	0	0	0	0	1	1	0	0
2	1	1	1	0	0	0	1	1	1	0	0
3	0	1	1	0	0	0	1	1	1	0	1
4	0	1	1	0	0	0	1	0	1	0	1

TABLE TIME2(K,I)

	1	2	3	4	5	6	7	8	9	10	11
1	1	0	4	1	5	6	0	0	3	5	0
2	1	0	4	1	5	6	2	0	3	5	0
3	0	0	4	1	5	6	2	0	3	5	3
4	0	0	4	0	0	6	2	0	3	5	3

TABLE TIME4(K,I)

	1	2	3	4	5	6	7	8	9	10	11
1	1	0	1	1	1	1	0	0	1	1	0
2	1	0	1	1	1	1	1	0	1	1	0
3	0	0	1	1	1	1	1	0	1	1	1
4	0	0	1	0	0	1	1	0	1	1	1

VARIABLES V(I,K),X1(K),X2(K),B(K),Z

BINARY VARIABLES V,X1,X2,B;

EQUATIONS AS(I)

CYC1(K)

CYC2(K)

ST1(K)

ST2(K)

OBJ

D1

D2

D3

D4

PR1

PR2

PR3

PR4

PR5

PR6

PR7

PR8

PR9

PR10

PR11

PR12

PR13;

AS(I).. SUM(K, ASS(I,K)*V(I,K)) =E= 1;
CYC1(K).. SUM(I, TIME1(I,K)*V(I,K)) =L=10;
CYC2(K).. SUM(I, TIME2(I,K)*V(I,K)) =L=10;
ST1(K).. SUM(I, TIME3(I,K)*V(I,K))-SAYIA(K)*X1(K) =L=0;
ST2(K).. SUM(I, TIME4(I,K)*V(I,K))-SAYIB(K)*X2(K) =L=0;
OBJ.. Z =E= SUM(K, B(K));
D1.. X1("1")+X2("1")-2*B("1") =E= 0;
D2.. X1("2")+X2("2")-2*B("2") =E= 0;
D3.. X1("3")+X2("3")-2*B("3") =E= 0;
D4.. X1("4")+X2("4")-2*B("4") =E= 0;
PR1.. SUM(K,ASS("1",K)*KAT(K)*V("1",K))-
SUM(K,ASS("2",K)*KAT(K)*V("2",K))=L=0;
PR2.. SUM(K,ASS("1",K)*KAT(K)*V("1",K))-
SUM(K,ASS("3",K)*KAT(K)*V("3",K))=L=0;
PR3.. SUM(K,ASS("1",K)*KAT(K)*V("1",K))-
SUM(K,ASS("4",K)*KAT(K)*V("4",K))=L=0;
PR4.. SUM(K,ASS("1",K)*KAT(K)*V("1",K))-
SUM(K,ASS("8",K)*KAT(K)*V("8",K))=L=0;
PR5.. SUM(K,ASS("2",K)*KAT(K)*V("2",K))-
SUM(K,ASS("7",K)*KAT(K)*V("7",K))=L=0;
PR6.. SUM(K,ASS("3",K)*KAT(K)*V("3",K))-
SUM(K,ASS("7",K)*KAT(K)*V("7",K))=L=0;
PR7.. SUM(K,ASS("4",K)*KAT(K)*V("4",K))-
SUM(K,ASS("5",K)*KAT(K)*V("5",K))=L=0;
PR8.. SUM(K,ASS("5",K)*KAT(K)*V("5",K))-
SUM(K,ASS("6",K)*KAT(K)*V("6",K))=L=0;
PR9.. SUM(K,ASS("6",K)*KAT(K)*V("6",K))-
SUM(K,ASS("7",K)*KAT(K)*V("7",K))=L=0;
PR10.. SUM(K,ASS("7",K)*KAT(K)*V("7",K))-

```

SUM(K,ASS("11",K)*KAT(K)*V("11",K))=L=0;
PR11.. SUM(K,ASS("8",K)*KAT(K)*V("8",K))-
SUM(K,ASS("9",K)*KAT(K)*V("9",K))=L=0;
PR12.. SUM(K,ASS("9",K)*KAT(K)*V("9",K))-
SUM(K,ASS("10",K)*KAT(K)*V("10",K))=L=0;
PR13.. SUM(K,ASS("10",K)*KAT(K)*V("10",K))-
SUM(K,ASS("11",K)*KAT(K)*V("11",K))=L=0;
MODEL DEN5 /ALL/;
SOLVE DEN5 USING MIP MINIMIZING Z;

```



EK-5

Karışık Modelli MHD Problemlerinde Herbir İstasyona Atanmaya
Aday Görevleri Belirleyen *Program WKM* 'nin FORTRAN Diliyle
Kodlanmış Yazılımı ve Örnek Bir Problem İçin Çözüm Çıktısı

Bu prosedür, GAMS çözümü için gerekli tabloların oluşturulmasını sağlamak için geliştirilmiştir. Bu program 4 ürünlü bir birleştirilmiş diyagramdaki her bir istasyona atanması muhtemel görevleri, ürün bazında ve bütün olarak ayrı ayrı vermektedir. Toplam düğüm sayısı ve geliştirilen sezgisel metottan (Program Mıxalb) elde edilen istasyon sayısı (KK), E_i L_i değerleri (Program EiLi) ve her bir ürün için görev zamanları, programa girdi olarak verilmektedir. Neticede Program, her bir istasyon için muhtemel görevleri ve zamanları ile bu görevlerin hangi ürünlere ait olduğunu vermektedir. Programı 4 ten fazla ürüne sahip problemler için de genişletmek mümkündür.

Program WKM

INTEGER E(60),L(60),W(60),TUR(60),N(60),WA(60)

INTEGER WB(60),WC(60),WD(60)

INTEGER TIME1(60),TIME2(60),TIME3(60),TIME4(60)

INTEGER S,SIR,SIRB,SIRC,SIRD

OPEN(1,FILE='INPUT.DAT')

OPEN(2,FILE='TERCIH.DAT')

OPEN(3,FILE='1URUN.REP')

OPEN(4,FILE='2URUN.REP')

OPEN(5,FILE='3URUN.REP')

OPEN(6,FILE='4URUN.REP')

S=0

SIR=0

SIRB=0

SIRC=0

SIRD=0

K=1

WRITE(*,91)

91 FORMAT(2X,'SEZGISELDEN ELDE EDILEN K DEGERI ?')

READ(*,*) KK

READ(1,*)IS

DO 120 I=1,IS


```

120 READ(1,*) TIME1(I),TIME2(I),TIME3(I),TIME4(I)
    DO 1111 I=1,IS
        WRITE(*,11) I,I
11  FORMAT(2X,'E(',I2,') VE L(',I2,') YI GIR:')
        READ(*,*) E(I)
        READ(*,*) L(I)
1111 CONTINUE
    DO 1 I=1,IS
        WRITE(*,13)I
13  FORMAT(2X,I2,'.DUGUM  HANGI URUNE AIT'/2X,27('*')/
        2X,'SADECE 1.URUN :1 SEC'/2X,'SADECE 2.URUN :2
        SEC'/2X,'SADECE 3.URUN :3 SEC'/2X,'SADECE 4.URUN :4
        SEC'/2X,'1-2-3 .URUN :5 SEC'/2X,'1-2-4.URUN :6 SEC'/
        2X,'1-3-4 .URUN :7 SEC'/2X,'2-3-4 .URUN :8 SEC'/
        2X,'1-2 .URUN :9 SEC'/2X,'1-3 .URUN :10 SEC'/
        2X,'1-4 .URUN :11 SEC'/2X,'2-3 .URUN :12 SEC'/
        2X,'2-4 .URUN :13 SEC'/2X,'3-4 .URUN :14 SEC'/
        2X,'1-2-3-4 .URUN :15 SEC'//)
        WRITE(*,37) I
37  FORMAT(2X,I2,' . DUGUM ICIN SECIMINIZ ?')
        READ(*,*) J
        TUR(I)=J
        WRITE(2,*) TUR(I)
1  CONTINUE
    CLOSE(2)
    OPEN(2,FILE='S**.DAT')
    DO 99 I=1,IS
99  READ(2,*) TUR(I)
10  DO 2 I=1,IS
        IF((E(I).LE.K).AND.(L(I).GE.K)) GOTO 5
        GOTO 2
5  S=S+1
        W(S)=I
        N(S)=TUR(I)
        IF(N(S).EQ.1.OR.N(S).EQ.5.OR.N(S).EQ.6.OR.N(S).EQ.7.OR.
        * N(S).EQ.9.OR.N(S).EQ.10.OR.N(S).EQ.11.OR.

```

```

      * N(S).EQ.15) GOTO 44
      GOTO 2
44   SIR=SIR+1
      WA(SIR)=I
      2 CONTINUE
      GOTO 60
71   S=0
      DO 22 I=1,IS
      IF((E(I).LE.K).AND.(L(I).GE.K)) GOTO 51
      GOTO 22
51   S=S+1
      W(S)=I
      N(S)=TUR(I)
      IF(N(S).EQ.2.OR.N(S).EQ.5.OR.N(S).EQ.6.OR.N(S).EQ.8.OR.
      * N(S).EQ.9.OR.N(S).EQ.12.OR.N(S).EQ.13.OR.
      * N(S).EQ.15) GOTO 45
      GOTO 22
45   SIRB=SIRB+1
      WB(SIRB)=I
      22 CONTINUE
      GOTO 61
72   S=0
      DO 23 I=1,IS
      IF((E(I).LE.K).AND.(L(I).GE.K)) GOTO 52
      GOTO 23
52   S=S+1
      W(S)=I
      N(S)=TUR(I)
      IF(N(S).EQ.3.OR.N(S).EQ.5.OR.N(S).EQ.7.OR.N(S).EQ.8.OR.
      * N(S).EQ.10.OR.N(S).EQ.12.OR.N(S).EQ.14.OR.
      * N(S).EQ.15) GOTO 46
      GOTO 23
46   SIRC=SIRC+1
      WC(SIRC)=I
      23 CONTINUE
      GOTO 62

```

```

73 S=0
    DO 24 I=1,IS
        IF((E(I).LE.K).AND.(L(I).GE.K)) GOTO 57
        GOTO 24
57 S=S+1
    W(S)=I
    N(S)=TUR(I)
    IF(N(S).EQ.4.OR.N(S).EQ.6.OR.N(S).EQ.7.OR.N(S).EQ.8.OR.
* N(S).EQ.11.OR.N(S).EQ.13.OR.N(S).EQ.14.OR.
* N(S).EQ.15) GOTO 48
    GOTO 24
48 SIRD=SIRD+1
    WD(SIRD)=I
24 CONTINUE
    GOTO 65
60 WRITE(*,12) K
12 FORMAT(2X,I2,'.ISTASYON ICIN TOPLAM ADAY ---SAHIP
* OLDUGU URUN'/)
    DO 3 I=1,S
        3 WRITE(*,14) W(I),N(I)
14 FORMAT(18X,I2,'----->',I2)
    PAUSE
    WRITE(*,*)'1. KUTUGE ISLENIYOR'
    WRITE(12,93)K
93 FORMAT(I2,'.STAT--1.URUN ICIN---')
    WRITE(3,*) (WA(I),I=1,SIR)
    WRITE(3,*) 'ZAMANI'
    WRITE(3,*) (TIME1(WA(I)),I=1,SIR)
    GOTO 71
61 WRITE(*,*)'2. KUTUGE ISLENIYOR'
    WRITE(4,94)K
94 FORMAT(I2,'.STAT--2.URUN ICIN--')
    WRITE(4,*) (WB(I),I=1,SIRB)
    WRITE(4,*) 'ZAMANI'
    WRITE(4,*) (TIME2(WB(I)),I=1,SIRB)
    GOTO 72

```

```
62 WRITE(*,*)'3. KUTUGE ISLENIYOR'
   WRITE(5,95)K
95  FORMAT(I3,'.STAT--3.URUN ICIN--')
   WRITE(5,*) (WC(I),I=1,SIRC)
   WRITE(5,*) 'ZAMANI'
   WRITE(5,*) (TIME3(WC(I)),I=1,SIRC)
   GOTO 73
65 WRITE(*,*)'4. KUTUGE ISLENIYOR'
   WRITE(6,96)K
96  FORMAT(I2,'.STAT--4.URUN ICIN--')
   WRITE(6,*) (WD(I),I=1,SIRD)
   WRITE(6,*) 'ZAMANI'
   WRITE(6,*) (TIME4(WD(I)),I=1,SIRD)
100 PAUSE
    K=K+1
    IF (K.GT.KK) GOTO 20
    S=0
    SIR=0
    SIRB=0
    SIRC=0
    SIRD=0
    GOTO 10
20  END
```

2 ürün, 8 istasyon için, Program WKM 'den elde edilen sonuçlar (Ürün bazında, her bir istasyon için muhtemel görevler ve zamanları) aşağıda verilmiştir:

1.STAT--1.URUN ICIN---

1	2	3	5
ZAMANLAR			
7	3	2	5

2.STAT--1.URUN ICIN---

1	2	3	5	6
7	8	9	11	12
ZAMANLAR				
7	3	2	5	1
5				
4	2	5	2	

3.STAT--1.URUN ICIN---

2	3	5	6	7
8				
9	11	12	14	15
ZAMANLAR				
3	2	5	1	5
4				
2	5	2	0	1

4.STAT--1.URUN ICIN---

2	6	7	8	9
11	12	14	15	16
18	19	20	21	22
23				
ZAMANLAR				
3	1	5	4	2
5	2	0	1	6
3	2	1	2	5
4				

5.STAT--1.URUN ICIN---

11	12	14	15	16
18	19	20	21	22

	23	24	26	27	
ZAMANLAR					
	5	2	0	1	6
	3	2	1	2	5
	4	5	2	4	
6. STAT--1. URUN ICIN--					
	16	18	19	20	21
	22	23	24	26	27
	28	29			
ZAMANLAR					
	6	3	2	1	2
	5	4	5	2	4
	3	2			
7. STAT--1. URUN ICIN--					
	19	20	21	22	23
	24	26	27	28	29
	30				
ZAMANLAR					
	2	1	2	5	4
	5	2	4	3	2
	6				
8. STAT--1. URUN ICIN--					
	27	28	29	30	
ZAMANLAR					
	4	3	2	6	

1. STAT--2. URUN ICIN--					
	1	2	4	5	
ZAMANLAR					
	7	3	2	2	
2. STAT--2. URUN ICIN--					
	1	2	4	5	6
	7	9	10	12	13
ZAMANLAR					
	7	3	2	2	1
	5	3	4	3	7

3. STAT--2.URUN ICIN--

2	4	5	6	7
9	10	12	13	14
15				

ZAMANLAR

3	2	2	1	5
3	4	3	7	0
2				

4. STAT--2.URUN ICIN--

2	6	7	9	10
12	13	14	15	16
17	18	20	21	23

ZAMANLAR

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

5. STAT--2.URUN ICIN--

10	12	13	14	15
16	17	18	20	21
23	24	25	26	27

ZAMANLAR

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

6. STAT--2.URUN ICIN--

16	17	18	20	21
23	24	25	26	27
28				

ZAMANLAR

6	5	3	3	3
3	4	1	3	4
3				

7. STAT--2.URUN ICIN--

17	20	21	23	24
25	26	27	28	30

ZAMANLAR

5	3	3	3	4
1	3	4	3	6

8.STAT--2.URUN ICIN--

27	28	30
----	----	----

ZAMANLAR

4	3	6
---	---	---



EK-6

0-1 Amaç Programlama Modeli ((STA) 1. Öncelikli Amaç)

Model 6.1

Amaç P_1 : İş istasyon sayısı bütün ürünler için 3 'den fazla olmamalıdır.

İkili olmayan değişkenler

STA : İstasyon sınırının pozitif sapma değişkeni.

CYC1 : 1. ürün çevrim zamanının pozitif sapma değişkeni.

CYC2 : 2. ürün çevrim zamanının pozitif sapma değişkeni.

BOLGE : Bölgeleme kısıtının negatif sapma değişkeni.

PTA_k : 1. ürünün k. istasyonu için çevrim zamanının negatif sapma değişkeni.

PTB_k : 2. ürünün k. istasyonu için çevrim zamanının negatif sapma değişkeni.

ISTNG : İş istasyon sınırının negatif sapma değişkeni.

BLG : Bölgeleme kısıtının pozitif sapma değişkeni.

Amaç fonksiyonu

min STA

st

Atama Kısıtları

$$V11+V12=1$$

$$V21+V22+V23+V24=1$$

$$V31+V32+V33+V34=1$$

$$V41+V42+V43=1$$

$$V51+V52+V53=1$$

$$V61+V62+V63+V64=1$$

$$V72+V73+V74=1$$

$$V81+V82+V83=1$$

$$V91+V92+V93+V94=1$$

$$V101+V102+V103+V104=1$$

$$V113+V114=1$$

Öncelik Kısıtları

$$V11+2V12-V31-2V32-3V33-4V34 \leq 0$$

$$V11+2V12-V41-2V42-3V43 \leq 0$$

$$V11+2V12-V21-2V22-3V23-4V24 \leq 0$$

$$V11+2V12-V81-2V82-3V83 \leq 0$$

$$V41+2V42+3V43-V51-2V52-3V53 \leq 0$$

$$V51+2V52+3V53-V61-2V62-3V63-4V64 \leq 0$$

$V31+2V32+3V33+4V34-2V72-3V73-4V74 \leq 0$
 $V61+2V62+3V63+4V64-2V72-3V73-4V74 \leq 0$
 $V21+2V22+3V23+4V24-2V72-3V73-4V74 \leq 0$
 $V81+2V82+3V83-V91-2V92-3V93-4V94 \leq 0$
 $V91+2V92+3V93+4V94-V101-2V102-3V103-4V104 \leq 0$
 $2V72+3V73+4V74-3V113-4V114 \leq 0$
 $V101+2V102+3V103+4V104-3V113-4V114 \leq 0$

Bölgeleme Kısıtları

$V91+2V92+3V93+4V94-V11-2V12+BOLGE-BLG=1$

Çevrim Zamanı Amaç Kısıtları (1.ürün için)

$V11+5V21+4V31+4V81+3V91+PTA1-CYC1=10$
 $V12+5V22+4V32+2V72+4V82+3V92+PTA2-CYC1=10$
 $5V23+4V33+2V73+4V83+3V93+3V113+PTA3-CYC1=10$
 $5V24+4V34+2V74+3V94+3V114+PTA4-CYC1=10$

Çevrim Zamanı Amaç Kısıtı (2.ürün için)

$V11+4V31+V41+5V51+6V61+3V91+5V101+PTB1-CYC2=8$
 $V12+4V32+V42+5V52+6V62+2V72+3V92+5V102+PTB2-CYC2=8$
 $4V33+V43+5V53+6V63+2V73+3V93+5V103+3V113+PTB3-CYC2=8$
 $4V34+6V64+2V74+3V94+5V104+3V114+PTB4-CYC2=8$

İş İstasyon Kısıtları

$V11+V21+V31+V81+V91-5X11 \leq 0$
 $V12+V22+V32+V72+V82+V92-6X21 \leq 0$
 $V23+V33+V73+V83+V93+V113-6X31 \leq 0$
 $V24+V34+V74+V94+V114-5X41 \leq 0$
 $V11+V31+V41+V51+V61+V91+V101-7X12 \leq 0$
 $V12+V32+V42+V52+V62+V72+V92+V102-8X22 \leq 0$
 $V33+V43+V53+V63+V73+V93+V103+V113-8X32 \leq 0$
 $V34+V64+V74+V94+V104+V114-6X42 \leq 0$

$X11+X12-2A1=0$

$X21+X22-2A2=0$

$X31+X32-2A3=0$

$X41+X42-2A4=0$

İş İstasyon Amaç Kısıtı

$A1+A2+A3+A4+ISTNG-STA=3$

End

Tüm X_{km} 0 veya 1

Tüm V_{ik} 0 veya 1

Tüm A_k 0 veya 1

Diğer tüm değişkenler ≥ 0



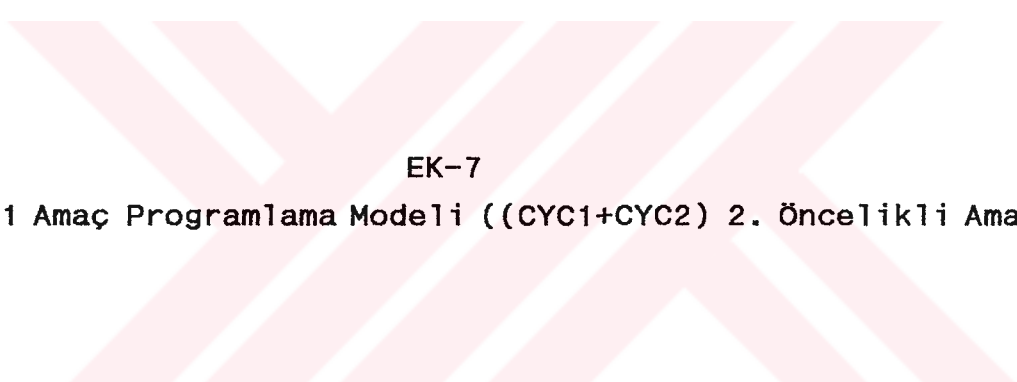
Model 6.1 'in çalıştırılması sonucu elde edilen 0-1 ve 0 olmayan değişken değerleri ile optimal amaç fonksiyon değeri (sapma değişken değeri) aşağıda verilmiştir.

OBJECTIVE FUNCTION VALUE

1) .000000000

VARIABLE	VALUE	REDUCED COST
X11	1.000000	.000000
X21	1.000000	.000000
X31	.000000	.000000
X41	1.000000	.000000
X12	1.000000	.000000
X22	1.000000	.000000
X32	.000000	.000000
X42	1.000000	.000000
A1	1.000000	.000000
A2	1.000000	.000000
A3	.000000	.000000
A4	1.000000	.000000
V11	1.000000	.000000
V12	.000000	.000000
V21	.000000	.000000
V22	1.000000	.000000
V23	.000000	.000000
V24	.000000	.000000
V31	.000000	.000000
V32	1.000000	.000000
V33	.000000	.000000
V34	.000000	.000000
V41	1.000000	.000000
V42	.000000	.000000
V43	.000000	.000000
V51	1.000000	.000000
V52	.000000	.000000

V53	.000000	.000000
V61	1.000000	.000000
V62	.000000	.000000
V63	.000000	.000000
V64	.000000	.000000
V72	.000000	.000000
V73	.000000	.000000
V74	1.000000	.000000
V81	1.000000	.000000
V82	.000000	.000000
V83	.000000	.000000
V91	.000000	.000000
V92	.000000	.000000
V93	.000000	.000000
V94	1.000000	.000000
V101	.000000	.000000
V102	.000000	.000000
V103	.000000	.000000
V104	1.000000	.000000
V113	.000000	.000000
V114	1.000000	.000000
STA	.000000	1.000000
BOLGE	.000000	.000000
BLG	2.000000	.000000
PTA1	5.000000	.000000
CYC1	.000000	.000000
PTA2	1.000000	.000000
PTA3	10.000000	.000000
PTA4	2.000000	.000000
PTB1	.000000	.000000
CYC2	5.000000	.000000
PTB2	9.000000	.000000
PTB3	13.000000	.000000
PTB4	.000000	.000000
ISTNG	.000000	.000000



EK-7

0-1 Amaç Programlama Modeli ((CYC1+CYC2) 2. Öncelikli Amaç)

Model 7.1

Amaç P_2 : Çevrim zamanı 1. ürün için 10 dk 'yı, 2. ürün için ise 8 dk 'yı geçmemelidir.

Amaç fonksiyonu

min $CYC1 + CYC2$

st

Atama Kısıtları

$$V11 + V12 = 1$$

$$V21 + V22 + V23 + V24 = 1$$

$$V31 + V32 + V33 + V34 = 1$$

$$V41 + V42 + V43 = 1$$

$$V51 + V52 + V53 = 1$$

$$V61 + V62 + V63 + V64 = 1$$

$$V72 + V73 + V74 = 1$$

$$V81 + V82 + V83 = 1$$

$$V91 + V92 + V93 + V94 = 1$$

$$V101 + V102 + V103 + V104 = 1$$

$$V113 + V114 = 1$$

Öncelik Kısıtları

$$V11 + 2V12 - V31 - 2V32 - 3V33 - 4V34 \leq 0$$

$$V11 + 2V12 - V41 - 2V42 - 3V43 \leq 0$$

$$V11 + 2V12 - V21 - 2V22 - 3V23 - 4V24 \leq 0$$

$$V11 + 2V12 - V81 - 2V82 - 3V83 \leq 0$$

$$V41 + 2V42 + 3V43 - V51 - 2V52 - 3V53 \leq 0$$

$$V51 + 2V52 + 3V53 - V61 - 2V62 - 3V63 - 4V64 \leq 0$$

$$V31 + 2V32 + 3V33 + 4V34 - 2V72 - 3V73 - 4V74 \leq 0$$

$$V61 + 2V62 + 3V63 + 4V64 - 2V72 - 3V73 - 4V74 \leq 0$$

$$V21 + 2V22 + 3V23 + 4V24 - 2V72 - 3V73 - 4V74 \leq 0$$

$$V81 + 2V82 + 3V83 - V91 - 2V92 - 3V93 - 4V94 \leq 0$$

$$V91 + 2V92 + 3V93 + 4V94 - V101 - 2V102 - 3V103 - 4V104 \leq 0$$

$$2V72 + 3V73 + 4V74 - 3V113 - 4V114 \leq 0$$

$$V101 + 2V102 + 3V103 + 4V104 - 3V113 - 4V114 \leq 0$$

Bölgeleme Amaç Kısıtı

$$V91 + 2V92 + 3V93 + 4V94 - V11 - 2V12 + BOLGE - BLG = 1$$

Çevrim Zamanı Amaç Kısıtı (1. Ürün için)

$$V11 + 5V21 + 4V31 + 4V81 + 3V91 + PTA1 - CYC1 = 10$$

$$V12+5V22+4V32+2V72+4V82+3V92+PTA2-CYC1=10$$

$$5V23+4V33+2V73+4V83+3V93+3V113+PTA3-CYC1=10$$

$$5V24+4V34+2V74+3V94+3V114+PTA4-CYC1=10$$

Çevrim Zamanı Amaç Kısıtı (2. ürün için)

$$V11+4V31+V41+5V51+6V61+3V91+5V101+PTB1-CYC2=8$$

$$V12+4V32+V42+5V52+6V62+2V72+3V92+5V102+PTB2-CYC2=8$$

$$4V33+V43+5V53+6V63+2V73+3V93+5V103+3V113+PTB3-CYC2=8$$

$$4V34+6V64+2V74+3V94+5V104+3V114+PTB4-CYC2=8$$

İş İstasyon Kısıtları

$$V11+V21+V31+V81+V91-5X11 \leq 0$$

$$V12+V22+V32+V72+V82+V92-6X21 \leq 0$$

$$V23+V33+V73+V83+V93+V113-6X31 \leq 0$$

$$V24+V34+V74+V94+V114-5X41 \leq 0$$

$$V11+V31+V41+V51+V61+V91+V101-7X12 \leq 0$$

$$V12+V32+V42+V52+V62+V72+V92+V102-8X22 \leq 0$$

$$V33+V43+V53+V63+V73+V93+V103+V113-8X32 \leq 0$$

$$V34+V64+V74+V94+V104+V114-6X42 \leq 0$$

$$X11+X12-2A1=0$$

$$X21+X22-2A2=0$$

$$X31+X32-2A3=0$$

$$X41+X42-2A4=0$$

İş İstasyon Amaç Kısıtı

$$A1+A2+A3+A4+ISTNG-STA=3$$

Model 6.1 'den elde edilen sapma değişken kısıtı

$$STA=0$$

End

Tüm X_{km} 0 veya 1

Tüm V_{ik} 0 veya 1

Tüm A_k 0 veya 1

Diğer tüm değişkenler ≥ 0

Model 7.1 'in çalıştırılması sonucu elde edilen 0-1 ve sıfır olmayan değişken değerleri ile amaç fonksiyon değeri (sapma değişken değeri) aşağıda verilmiştir.

OBJECTIVE FUNCTION VALUE

1) 2.00000000

VARIABLE	VALUE	REDUCED COST
X11	.000000	.000000
X21	1.000000	.000000
X31	1.000000	.000000
X41	1.000000	.000000
X12	.000000	.000000
X22	1.000000	.000000
X32	1.000000	.000000
X42	1.000000	.000000
A1	.000000	.000000
A2	1.000000	.000000
A3	1.000000	.000000
A4	1.000000	.000000
V11	.000000	.000000
V12	1.000000	.000000
V21	.000000	.000000
V22	.000000	.000000
V23	.000000	.000000
V24	1.000000	5.000000
V31	.000000	.000000
V32	.000000	.000000
V33	1.000000	.000000
V34	.000000	8.000000
V41	.000000	.000000
V42	1.000000	.000000
V43	.000000	.000000
V51	.000000	.000000
V52	1.000000	.000000

V53	.000000	.000000
V61	.000000	.000000
V62	.000000	.000000
V63	1.000000	.000000
V64	.000000	6.000000
V72	.000000	.000000
V73	.000000	.000000
V74	1.000000	4.000000
V81	.000000	.000000
V82	1.000000	.000000
V83	.000000	.000000
V91	.000000	.000000
V92	1.000000	.000000
V93	.000000	.000000
V94	.000000	6.000000
V101	.000000	.000000
V102	.000000	.000000
V103	.000000	.000000
V104	1.000000	5.000000
V113	.000000	.000000
V114	1.000000	6.000000
CYC1	.000000	.000000
CYC2	2.000000	.000000
BOLGE	1.000000	.000000
BLG	.000000	.000000
PTA1	10.000000	.000000
PTA2	2.000000	.000000
PTA3	6.000000	.000000
PTA4	.000000	1.000000
PTB1	10.000000	.000000
PTB2	.000000	.000000
PTB3	.000000	.000000
PTB4	.000000	1.000000
ISTNG	.000000	.000000
STA	.000000	.000000

EK-8
0-1 Amaç Programlama Modeli ((BOLGE) 3. Öncelikli Amaç).

Model 8.1

Amaç P_3 : Her iki ürün için de ortak 1 ve 9 nolu görevlerin aynı istasyonda yapılmasına izin verilmemelidir.

Amaç fonksiyonu

min BOLGE

st

Atama Kısıtları

$$V_{11}+V_{12}=1$$

$$V_{21}+V_{22}+V_{23}+V_{24}=1$$

$$V_{31}+V_{32}+V_{33}+V_{34}=1$$

$$V_{41}+V_{42}+V_{43}=1$$

$$V_{51}+V_{52}+V_{53}=1$$

$$V_{61}+V_{62}+V_{63}+V_{64}=1$$

$$V_{72}+V_{73}+V_{74}=1$$

$$V_{81}+V_{82}+V_{83}=1$$

$$V_{91}+V_{92}+V_{93}+V_{94}=1$$

$$V_{101}+V_{102}+V_{103}+V_{104}=1$$

$$V_{113}+V_{114}=1$$

Öncelik Kısıtları

$$V_{11}+2V_{12}-V_{31}-2V_{32}-3V_{33}-4V_{34}\leq 0$$

$$V_{11}+2V_{12}-V_{41}-2V_{42}-3V_{43}\leq 0$$

$$V_{11}+2V_{12}-V_{21}-2V_{22}-3V_{23}-4V_{24}\leq 0$$

$$V_{11}+2V_{12}-V_{81}-2V_{82}-3V_{83}\leq 0$$

$$V_{41}+2V_{42}+3V_{43}-V_{51}-2V_{52}-3V_{53}\leq 0$$

$$V_{51}+2V_{52}+3V_{53}-V_{61}-2V_{62}-3V_{63}-4V_{64}\leq 0$$

$$V_{31}+2V_{32}+3V_{33}+4V_{34}-2V_{72}-3V_{73}-4V_{74}\leq 0$$

$$V_{61}+2V_{62}+3V_{63}+4V_{64}-2V_{72}-3V_{73}-4V_{74}\leq 0$$

$$V_{21}+2V_{22}+3V_{23}+4V_{24}-2V_{72}-3V_{73}-4V_{74}\leq 0$$

$$V_{81}+2V_{82}+3V_{83}-V_{91}-2V_{92}-3V_{93}-4V_{94}\leq 0$$

$$V_{91}+2V_{92}+3V_{93}+4V_{94}-V_{101}-2V_{102}-3V_{103}-4V_{104}\leq 0$$

$$2V_{72}+3V_{73}+4V_{74}-3V_{113}-4V_{114}\leq 0$$

$$V_{101}+2V_{102}+3V_{103}+4V_{104}-3V_{113}-4V_{114}\leq 0$$

Bölgeleme Amaç Kısıtı

$$V_{91}+2V_{92}+3V_{93}+4V_{94}-V_{11}-2V_{12}+BOLGE-BLG=1$$

Çevrim Zamanı Amaç Kısıtı (1. ürün için)

$$V11+5V21+4V31+4V81+3V91+PTA1-CYC1=10$$

$$V12+5V22+4V32+2V72+4V82+3V92+PTA2-CYC1=10$$

$$5V23+4V33+2V73+4V83+3V93+3V113+PTA3-CYC1=10$$

$$5V24+4V34+2V74+3V94+3V114+PTA4-CYC1=10$$

Çevrim Zamanı Amaç Kısıtı (2. ürün için)

$$V11+4V31+V41+5V51+6V61+3V91+5V101+PTB1-CYC2=8$$

$$V12+4V32+V42+5V52+6V62+2V72+3V92+5V102+PTB2-CYC2=8$$

$$4V33+V43+5V53+6V63+2V73+3V93+5V103+3V113+PTB3-CYC2=8$$

$$4V34+6V64+2V74+3V94+5V104+3V114+PTB4-CYC2=8$$

İş İstasyon Kısıtları

$$V11+V21+V31+V81+V91-5X11 \leq 0$$

$$V12+V22+V32+V72+V82+V92-6X21 \leq 0$$

$$V23+V33+V73+V83+V93+V113-6X31 \leq 0$$

$$V24+V34+V74+V94+V114-5X41 \leq 0$$

$$V11+V31+V41+V51+V61+V91+V101-7X12 \leq 0$$

$$V12+V32+V42+V52+V62+V72+V92+V102-8X22 \leq 0$$

$$V33+V43+V53+V63+V73+V93+V103+V113-8X32 \leq 0$$

$$V34+V64+V74+V94+V104+V114-6X42 \leq 0$$

$$X11+X12-2A1=0$$

$$X21+X22-2A2=0$$

$$X31+X32-2A3=0$$

$$X41+X42-2A4=0$$

İş İstasyon Amaç Kısıtı

$$A1+A2+A3+A4+ISTNG-STA=3$$

Model 6.1 'den elde edilen sapma değ. Kısıtı

$$STA=0$$

Model 7.1 'den elde edilen sapma değ. Kısıtı

$$CYC1+CYC2=2$$

End

Tüm X_{km} 0 veya 1

Tüm V_{ik} 0 veya 1

Tüm A_k 0 veya 1

Diğer tüm değişkenler ≥ 0

Model 8.1 'in çalıştırılması sonucu elde edilen 0-1 ve 0 olmayan değişken değerleri ile amaç fonksiyon değeri, aşağıda verilmiştir.

OBJECTIVE FUNCTION VALUE

1) 1.00000000

VARIABLE	VALUE	REDUCED COST
X11	1.000000	.000000
X21	1.000000	.000000
X31	.000000	.000000
X41	1.000000	.000000
X12	1.000000	.000000
X22	1.000000	.000000
X32	.000000	.000000
X42	1.000000	.000000
A1	1.000000	.000000
A2	1.000000	.000000
A3	.000000	.000000
A4	1.000000	.000000
V11	1.000000	1.000000
V12	.000000	2.000000
V21	.000000	.000000
V22	1.000000	.000000
V23	.000000	.000000
V24	.000000	.000000
V31	.000000	.000000
V32	1.000000	.000000
V33	.000000	.000000
V34	.000000	.000000
V41	1.000000	.000000
V42	.000000	.000000
V43	.000000	.000000
V51	1.000000	.000000
V52	.000000	.000000

V53	.000000	.000000
V61	.000000	.000000
V62	1.000000	.000000
V63	.000000	.000000
V64	.000000	.000000
V72	.000000	.000000
V73	.000000	.000000
V74	1.000000	.000000
V81	1.000000	.000000
V82	.000000	.000000
V83	.000000	.000000
V91	1.000000	-1.000000
V92	.000000	-2.000000
V93	.000000	-3.000000
V94	.000000	-4.000000
V101	.000000	.000000
V102	.000000	.000000
V103	.000000	.000000
V104	1.000000	.000000
V113	.000000	.000000
V114	1.000000	.000000
BOLGE	1.000000	.000000
BLG	.000000	1.000000
PTA1	2.000000	.000000
CYC1	.000000	.000000
PTA2	1.000000	.000000
PTA3	10.000000	.000000
PTA4	5.000000	.000000
PTB1	.000000	.000000
CYC2	2.000000	.000000
PTB2	.000000	.000000
PTB3	10.000000	.000000
PTB4	.000000	.000000
ISTNG	.000000	.000000
STA	.000000	.000000

ÖZGEÇMİŞ

Hadi GÖKÇEN, 1964 yılında Ankara 'da doğdu. İlk orta ve lise eğitimini tamamladıktan sonra, 1982 yılında Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Endüstri Mühendisliği Bölümüne başladı ve 1987 yılında mezun oldu. 1989 yılında Gazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Ana Bilim Dalından Yüksek Lisans Derecesini aldı. Halen aynı Üniversitenin Endüstri Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır.