

ROBOTİK HÜCRELERDE HAT DENGEME

SUMRU ÜSTÜNOL

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ARALIK 2012

ANKARA

Fen Bilimleri Enstitü onayı

Prof. Dr. Ünver KAYNAK
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Prof. Dr. Ömer SAATÇIOĞLU
Anabilim Dalı Başkanı

SUMRU ÜSTÜNOL tarafından hazırlanan ROBOTİK HÜCRELERDE HAT DENGEME adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Hakan GÜLTEKİN
Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd. Doç. Dr. Nilgün FESCİOĞLU ÜNVER _____

Üye : Yrd. Doç. Dr. Hakan GÜLTEKİN _____

Üye : Yrd. Doç. Dr. Tansel ÖZYER _____

TEZ BİLDİRİMİ

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

Sumru ÜSTÜNOL

Üniversitesi : TOBB Ekonomi ve Teknoloji Üniversitesi
Enstitüsü : Fen Bilimleri
Anabilim Dalı : Endüstri Mühendisliği
Tez Danışmanı : Yrd. Doç. Dr. Hakan GÜLTEKİN
Tez Türü ve Tarihi : Yüksek Lisans – Aralık 2012

Sumru ÜSTÜNOL

ROBOTİK HÜCRELERDE HAT DENGEME

ÖZET

Bu çalışmada m makineli ve tek tip parça üreten bir robotik hücre ele alınmıştır. Makineler arası malzeme taşıma ve makinelerin yükleme boşaltma işlemleri robotlar tarafından gerçekleştirilmektedir. Bir parçanın üretimi için pek çok işlemden geçmesi gerekebilir. Bu tip sistemlerdeki makineler doğaları gereği CNC makineleridir ve bu makineler yüksek esnekliğe sahiptir. Bu esneklik tiplerinden bir tanesi de makinelerin değişik işlemleri yapabilme kabiliyetleridir. Gerekli kesici uçlar makinelerin uç haznelerinde yüklü olduğu sürece bu makineler değişik işlemleri yapabilmektedirler. Bu nedenle işlemlerin farklı makinelerde gerçekleştirilmesi mümkündür. Makine işlem zamanları bu atamaların sonucu olarak ortaya çıkmaktadır. Bunun yanı sıra işlemler arasında öncelik ilişkileri de bulunmaktadır. Amaç, çevrim zamanını en küçükleyecek şekilde işlemlerin makinelere atanmasını sağlamak ve en iyi 1-birim döngüsünü bulmaktır. Çevrim zamanı, uzun dönemde ardışık iki parçanın sistemden ortalama çıkış zamanıdır. Bu çalışmada robotun bir başlangıç durumundan başlayıp her makineyi bir kere yüklediği ve bir kere boşalttığı 1-birim robotik döngüler ele alınmıştır. NP-Zor olan bu problem için matematiksel model tabanlı bir kesin çözüm yöntemi ve bir sezgisel algoritma geliştirilmiştir. Sezgisel algoritmanın etkinliği bir deneysel çalışma ile test edilmiştir.

Anahtar Kelimeler: robotik hücreler, montaj hattı dengeleme, çizelgeleme, esnek üretim sistemleri.

University : **TOBB University of Economics and Technology**
Institute : **Institute of Natural and Applied Sciences**
Science Programme : **Industrial Engineering**
Supervisor : **Asst. Prof. Hakan GÜLTEKİN**
Degree Awarded and Date : **M.Sc. – DECEMBER 2012**

Sumru ÜSTÜNOL

LINE BALANCING IN ROBOTIC CELLS

ABSTRACT

In this study, an m-machine robotic cell producing a single part type is considered. The loading and unloading of the machines and the material transportation between the machines are carried out by a single robot. It is assumed that a part requires a number of operations to be processed on the machines, which are highly flexible CNC machines. One of the flexibilities that these machines possess is the capability of performing different operations as long as the required cutting tools are loaded on their tool magazines. Therefore, it is possible to process a task on one of several different alternative machines. The processing time of the part on a machine is the sum of the individual processing times of the operations assigned to that machine. There are precedence relations between the operations. The aim is to determine the assignment of the operations to the machines and the 1-unit robot move cycle that minimize the cycle time, long run average time between the production of two consecutive parts. In this study, 1-unit cycles in which starting with an initial state, the robot loads and unloads each machine exactly once and returns back to the initial state, are considered. A mathematical programming based exact solution procedure as well as a heuristic algorithm is developed for this NP-Hard problem. The effectiveness of the heuristic algorithm is tested through an extensive computational study.

Keywords: robotic cells, assembly line balancing, scheduling, flexible manufacturing systems.

TEŞEKKÜR

Yüksek lisans eğitimim boyunca bana rehberlik eden ve sevdiğim bir alanda araştırma yapma imkanı sağlayan, değerli katkıları ile çalışmamı yönlendiren danışmanım Hakan Gültekin'e sonsuz teşekkürlerimi sunarım.

Başta tezimi okuyarak tavsiyelerde bulunan Nilgün Fescioğlu Ünver ve Tansel Özyer'e, ayrıca yüksek lisans sürecinde derslerine katıldığım bütün hocalara teşekkürü bir borç bilirim.

Benden hiç bir zaman yardımlarını esirgemeyen, içten dostlukları ile her zaman destek olan Hande, Gizem, Şebnem, Tunç, Çağla, Arzu ve Tuba'ya, ayrıca tez problemimi defalarca anlatarak rahatsız ettiğim Cumhuriyet'e, tezimle alakalı en çok sorular soran ve bitince benim kadar sevinen Cansu'ya ve tezin kalbini uzun tutmalısın diyerek ve karamsarlığa düştüğümde hep yanımda olarak beni cesaretlerinden hatta ödüllendiren Hakan'a her zaman yanımda oldukları için sonsuz teşekkürler.

Maddi ve manevi her zaman desteklerini esirgemeyen, yanımda olduklarını hep hissettiğim aileme, özellikle anneme, babama ve çatışmalarımız sonucu aydınlanmamı sağlayan kardeşime minnettarım.

İçindekiler

1 Giriş	1
2 Literatür Taraması	4
2.1 Robotik Hücre Çizelgeleme	5
2.1.1 Tek Tutuculu Robotik Hücrelerde Çizelgeleme	7
2.1.2 Çift Tutuculu Robotik Hücrelerde Çizelgeleme	10
2.1.3 Çift Kollu Robotik Hücreler	11
2.2 Hat Dengeleme	12
2.2.1 Basit montaj hattı dengeleme	12
2.2.2 Genel Montaj Hattı Dengeleme	15
2.3 Özet	18
3 Problem Tanımı ve Eniyileme Modeli	20
3.1 Temel Kavramlar ve Problem Tanımı	20
3.2 Kesin Çözüm Yöntemi	23
4 Sezgisel Çözüm Yöntemi	28

4.1	Crama-van de Klundert [19] Algoritması	28
4.2	Sezgisel Yöntem	29
4.2.1	α , β ve γ Parametrelerinin Belirlenmesi	38
5	Deneysel Çalışma	42
5.1	Deneysel Çalışma Parametreleri	42
5.2	Sonuçların Analizi	47
6	Sonuç ve Değerlendirme	52
	KAYNAKLAR	55
	EKLER	63
A	4, 5 ve 6 Makineli Robotik Hücreler İçin Çevrim Zamanları	64
B	Deneysel Çalışma Sonuçları	75
	ÖZGEÇMİŞ	94

Şekil Listesi

2.1	Robotik hücreler sınıflandırması [28]	6
3.1	m makineli robotik hücre	21
3.2	Piramitsel permutasyonun örnek gösterimi: $A_0A_2A_4A_3A_1$ döngüsü ve Piramitsel permutasyona örnek olmayan gösterim: $A_0A_3A_2A_4A_1$ döngüsü	24
4.1	Sinüs grafiğine göre işlem zamanları	33
4.2	Çözüm süresi-hata ödünleşimi	39
5.1	Atanabilirlik kuralları doğrultusunda atanabilirlik matrisinin oluş- turulması	45

Tablo Listesi

4.1	Örnek 1'deki problem için işlem zamanları	35
4.2	Örnek 1' deki problem işlemlerinin öncelik ilişkileri b_{kj}	36
4.3	Örnek probleme ait atanabilirlik matrisi a_{ij}	36
4.4	Örnek 1' de ROHDA'ya göre oluşan makine işlem zamanları	37
4.5	Örnek 1' de GAMS' e göre oluşan makine işlem zamanları	38
4.6	α , β ve γ Parametrelerinin değişik değerleri ile elde edilen sonuçlar	40
4.7	α , β ve γ değerlerinin değişimi	41
5.1	Problem İsimleri ve İşlem Sayıları	43
5.2	Bütün parametreler ve kullanılan değerler	46
5.3	Makine sayısına göre	47
5.4	Problem Sayısına Göre	49
5.5	ϵ ve δ değerlerine göre	50
5.6	F-ratio değerlerine göre	51
B.1	4 makine $F - ratio = 0$ için ayrıntılı sonuçlar (1-6)	76
B.2	4 makine $F - ratio = 0$ için ayrıntılı sonuçlar (7-12)	77

B.3	4 makine $F - ratio = 0$ için ayrıntılı sonuçlar (13-17)	78
B.4	4 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (1-6)	79
B.5	4 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (7-12)	80
B.6	4 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (13-17)	81
B.7	5 makine $F - ratio = 0$ için ayrıntılı sonuçlar (1-6)	82
B.8	5 makine $F - ratio = 0$ için ayrıntılı sonuçlar (7-12)	83
B.9	5 makine $F - ratio = 0$ için ayrıntılı sonuçlar (13-17)	84
B.10	5 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (1-6)	85
B.11	5 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (7-12)	86
B.12	5 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (13-17)	87
B.13	6 makine $F - ratio = 0$ için ayrıntılı sonuçlar (1-6)	88
B.14	6 makine $F - ratio = 0$ için ayrıntılı sonuçlar (7-12)	89
B.15	6 makine $F - ratio = 0$ için ayrıntılı sonuçlar (13-17)	90
B.16	6 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (1-6)	91
B.17	6 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (7-12)	92
B.18	6 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (13-17)	93

1. Giriş

Teknolojide meydana gelen gelişmeler, müşteri ihtiyaçlarının zamanında giderilmesi ve beklentilerinin tam anlamıyla karşılanması gibi unsurlar sebebiyle işletmeler üretim süreçlerini sürekli olarak geliştirme ihtiyacı duymaktadır. Rekabet koşulları da göz önüne alındığında işletmeler kalitenin yanı sıra üretim süreçlerinde esneklik ve üretim hızı gibi ürün üretim aşamalarına da önem vermektedirler. Bu sebeplerle beraber üretimde daha düşük taşıma maliyetlerine olan ihtiyaç, en doğru ve en uygun malzeme taşıma sistemi seçimini önemli kılmıştır. Tüm üretim maliyetleri içerisinde malzeme taşıma maliyetleri oranının göz ardı edilemeyecek bir boyutta olduğu düşünüldüğünde, doğru seçimin önemi daha iyi ortaya çıkmaktadır.

Gelişen teknoloji ile beraber malzeme taşıma sistemlerinde de robotlar kullanılmaya başlanmıştır. Endüstride robotların temel hedefleri arasında, sistem maliyeti azalımı, kalite güvencesi, esnek üretim ve insana saygı gibi unsurlar yer almaktadır. Ayrıca robotların kullanılmasıyla işçilerin tehlikeli işlerde çalışması engellenmiş, malzeme taşıma gibi monoton ve bazen de ciddi oranda güç gerektiren işlerde de robotlar kullanılarak işçilerin farklı işlere yönlendirilmesi sağlanmıştır. Malzeme taşıma maliyetlerinin yanı sıra, düşük kapasite kullanımı ve verimsiz çalışmanın maliyeti de oldukça yüksektir. Bu sebeple montaj hatlarının verimli ve düzenli çalışabilmesi için, her iş istasyonuna dengeli miktarda iş dağıtımı yapılmalıdır. Bu da, hat dengeleme problemini ortaya çıkarmaktadır. Hat dengeleme problemi, hatta yerine getirilecek görevlerin, aralarındaki öncelik ilişkileri ve sisteme özgü kısıtlar ihlal edilmeden, belirli bir performans ölçütü en iyilenecek şekilde iş istasyonlarına atanması problemi. En iyilenecek performans ölçütleri arasında verilen makine sayısını aşmadan çevrim zamanının en küçüklenmesi veya verilen çevrim zamanını aşmadan kullanılacak iş istasyonu sayısının en küçüklenmesi

sayılabilir.

Bu çalışmada tek tip parça üreten bir robotik üretim hattı ele alınmaktadır. Bu üretim hattında makineler arası parça transferi ve makinelerin yüklenip boşaltılmaları bir endüstriyel robot tarafından yapılmaktadır. Bu tip sistemlerdeki makineler doğaları gereği CNC makineleridir ve bu makineler yüksek esnekliğe sahiptir. Bu esneklik tiplerinden bir tanesi de makinelerin değişik işlemleri yapabilme kabiliyetleridir. Gerekli kesici uçlar makinelerin uç haznelerinde yüklü olduğu sürece bu makineler değişik işlemleri yapabilmektedirler. Dolayısıyla, parçaların ihtiyaç duyduğu işlemler farklı makinelerde yapılabilmektedir ve hangi makinede yapılacağı bir karar değişkenidir. İşlemlerin makinelere atanmasına göre de parçaların makinelerdeki toplam işlem zamanları belirlenmiş olur. Diğer taraftan bu işlemler arasında öncelik ilişkileri bulunabilmektedir. Yani, bir işlemin işlenmeyi başarabilmesi için bazı başka işlemlerin tamamlanmış olması gerekir.

Robotlar belirli bir bilgisayar kodunu takip ettikleri için periyodik olarak aynı aktiviteleri tekrar etmek durumundadırlar. Bunun doğal bir sonucu olarak bu tip sistemlerde döngüsel çizelgeleme problemleri kullanılmaktadır. Robotun bir başlangıç durumundan başlayıp belirli aktiviteleri tekrar ettikten sonra başlangıç durumuna geri döndüğü çizelgelere döngüsel çizelgeleme denilmektedir. Robot, bu döngüyü sürekli tekrarlar. Bir döngü süresince sistemden çıkan parça sayısına göre döngüler isimlendirilirler. Buna göre, n -birim döngüleri, bir döngü içerisinde n tane parçanın üretildiği döngülerdir. Basitlikleri, kolay anlaşılabilirlikleri ve çok iyi sonuçlar vermelerinden dolayı literatürde en çok ele alınan döngüler 1-birim döngüleridir. Bu çalışmada da 1-birim döngüler ele alınmıştır. Ele alınan problemdeki amaç; üretim hızını en büyükmek için, öncelik ve atanabilirlik kısıtlarını sağlayacak şekilde işlemlerin makinelere atanmasını ve en iyi 1-birim robot hareket döngüsünü belirlemektir. Üretim hızının en büyükmesi çevrim zamanının en küçükmesine eşdeğerdir. Çevrim zamanı uzun vadede, sistemden çıkan ardışık iki parça arasında geçen ortalama zamandır. Ele alınan sistemde ise, robotun başlangıç durumundan başlayıp bu duruma tekrar dönene kadar geçirdiği zamana eşittir. Sistemin durumunu ise sistemdeki makinelerin doluluk/boşluk durumu ve robotun konumu (hangi makinenin önünde olduğu) şeklinde gösterilebilir.

Sadece 1-birim döngülerin ele alınması kısmi bir birleştirme sağlasa da, m

makinede $m!$ tane 1-birim döngüsü olmasından dolayı [72] optimal çözüme ulaşmak zordur. Ayrıca, hem en iyi 1-birim döngüsünü ve hem de işlemlerin makinelere atanmasını aynı anda belirleyecek tek bir matematiksel model kurulamaması problemin çözümünü karmaşıktır. Problemi optimal olarak çözebilmek için her 1-birim döngüsü için ayrı bir hat dengeleme (işlem atama) problemi çözdürülmelidir. Dolayısıyla m makineli bir sistemde $m!$ tane model çözdürülmelidir. Diğer taraftan, çözdürülecek $m!$ matematiksel modelin bir kısmı çok basit olsa da büyük bir kısmı NP-Zor'dur. Bu da sezgisel bir algoritmanın gerekliliğini ortaya çıkarmaktadır.

Bir sonraki bölümde ele alınan konuyla doğrudan ilintili olan robotik hücre çizelgeleme ve montaj hattı dengeleme literatüründe daha önce ele alınmış problemler ve elde edilen sonuçlar tarandıktan sonra Bölüm 3'te problemin detaylı tanımı yapılacak ve kurulan matematiksel modeller anlatılacaktır. Bölüm 4'te geliştirilen sezgisel algoritmanın detayları açıklanacak, Bölüm 5'te ise algoritmanın etkinliğini test etmek için yapılan deneysel çalışmanın sonuçları aktarılacaktır. Bölüm 6 da ise sonuçlar ve gelecek çalışma konuları anlatılacaktır.

2. Literatür Taraması

Modern üretim sistemlerinin kullanımının artması ile paralel bir şekilde robotik hücrelerin kullanımı da giderek artmaktadır. Bu artışa paralel olarak, bu konuda 1980lerde başlayan çalışmalar gün geçtikçe artmakta ve ihtiyaca yönelik olarak çeşitlenmektedir. Çalışmalar ilk olarak robotun tek kolunun bulunduğu ve bu kolda tek bir parça taşıyabildiği tek tutuculu robotik hücreler ve aynı tip parça üretimini ifade eden **tek tip** parça üretimi için ortaya çıksa da, gelişen teknoloji ve müşteri istekleri doğrultusunda farklı türde parçaların üretilbildiği **değişik tip** parça üretimi ve robotun ve üretim hücresinin değişik konfigürasyon ve teknolojik özelliklere sahip olduğu çok çeşitli problemler ele alınmıştır. Bu bölümde robotik hücre çizelgeleme literatüründeki çalışmalar detaylı bir şekilde incelenecektir.

Çalışmanın yakından ilgili olduğu bir diğer başlık ise "montaj hattı dengeleme"dir. Kalite ve verimliliğin artırılması için üretimde iyileştirilmenin yapılması gerekmektedir. Üretim hatlarının verimliliğinin artırılması için yapılan çalışmalardan biri olan montaj hattı dengeleme, bu çalışmalar arasında en sık karşılaşılanıdır. Montaj hattı dengeleme üzerine yapılan ilk çalışmalar 1950lere ortaya çıkmıştır. Literatürde hat dengeleme problemi üzerine pek çok yöntem bulunmaktadır. Bu çeşitlilik için montaj hatlarının karmaşıklığına göre bir sınıflandırma yapılmış, basit ve genel olmak üzere hatlar ikiye ayrılmıştır [10]. Problem çözümünde kullanılan yöntemlere göre ise matematiksel ve sezgisel olmak üzere iki başlık altında incelenecektir.

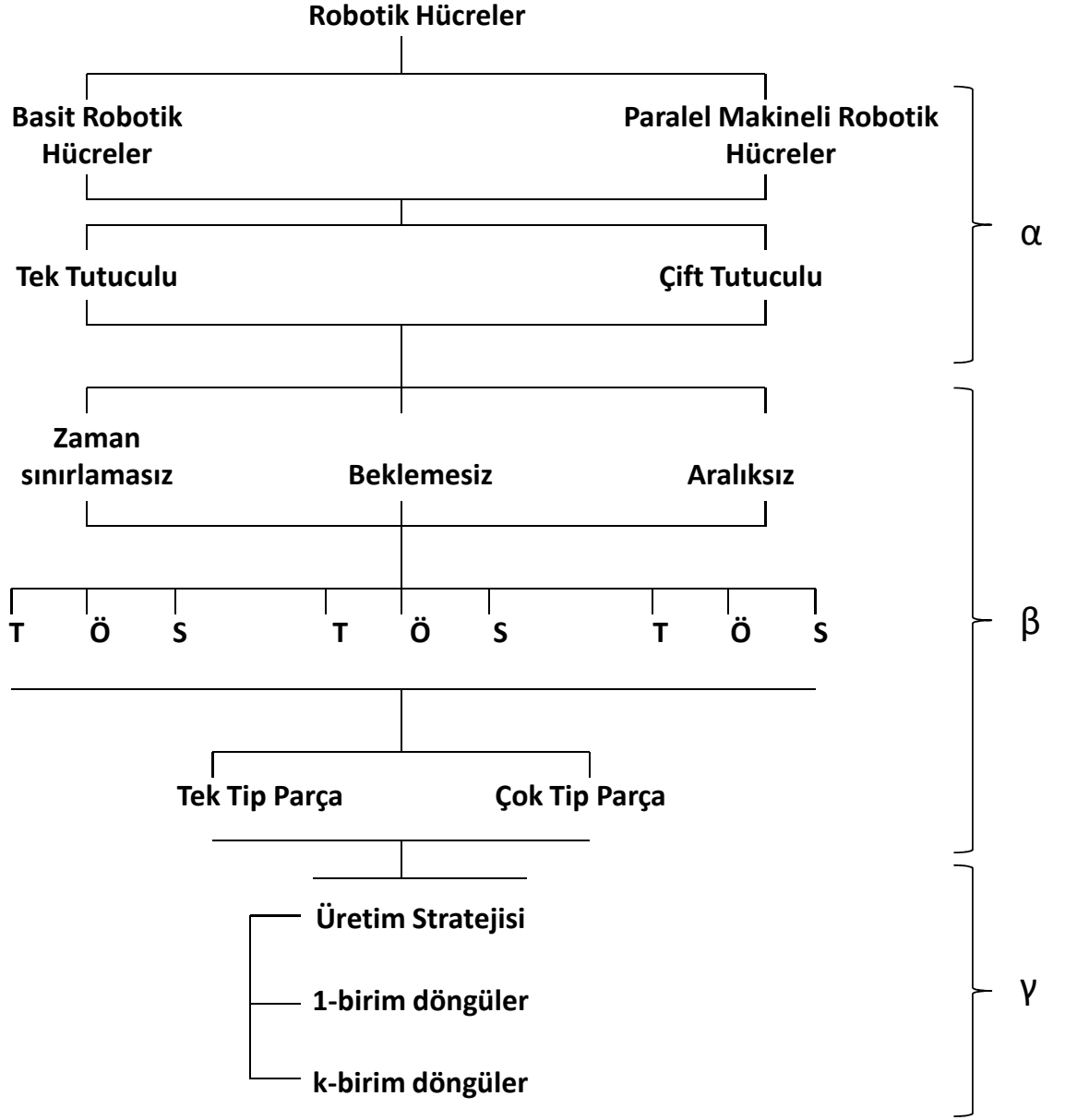
Bu bölümün geri kalan kısmında öncelikle robotik hücre çizelgeleme literatürü incelenecektir. İkinci kısımda ise montaj hattı dengeleme literatüründeki çalışmalar özellikle de mevcut çalışma ile ilgili olanlar detaylı bir şekilde incelenecektir.

2.1 Robotik Hücre Çizelgeleme

Gelişen teknolojiyle beraber ortaya çıkan robotik hücre çizelgeleme literatürü, günümüzden yaklaşık 20 yıl kadar önce Sethi vd.[72]'nin yapmış olduğu çalışma ile başlar. Teknolojinin sunduğu yeniliklerle beraber robotik hücreler de her geçen gün yenilenmektedir. Robotların tek seferde tek bir parçayı taşıyabildiği klasik "tek tutuculu" robotların yanı sıra, esnekliği ve üretim hızını artırmak amacıyla değişik özelliklere sahip robotlar da kullanılmaktadır. Bunlar arasında farklı yönlere dönük iki kolun farklı makinelerde aynı anda yükleme boşaltma yapabildiği "çift kollu robotlar" ve tek kola sahip olmasına rağmen aynı anda iki parça tutabilen ve taşıyabilen "çift tutuculu robotlar" sayılabilir. Tek tip parça ya da çok tip parça üretimi doğrultusunda da çalışmalar çeşitlilik göstermiştir. Yapılan çalışmalarda genellikle kolay anlaşılması, işlemlerin kolay yapılması ve optimal sonucu vermesi ya da optimale yakın sonuçlar vermesi sebebiyle 1-birim döngüler, daha genel ama analizi çok daha karmaşık olan k-birim döngülere göre daha çok kullanılmıştır. Geismar vd. [28], Crama vd. [18]'nin yapmış olduğu literatür taramaları sınıflandırma hakkında geniş bilgiler vermektedir. Bu çalışmalarda ayrıca, geniş bir literatür taraması da bulunmaktadır.

Geismar vd. [28]'nin yapmış oldukları sınıflandırma robotik hücreleri 3 bölgeye ayırır: α , β ve γ (Şekil 2.1). α makine çevresini ifade eder. Makinelerin dizilimine, robot sayısına ve robotların bulundurduğu tutucu sayısına göre bir ayırım söz konusudur. β süreç karakteristiklerini temsil eder. Birçok çalışmada makineler arasında herhangi bir stok alanının olmadığı varsayılmaktadır. Bunun doğal bir sonucu olarak da M_i makinesinde işlemi son bulmuş herhangi bir işlem M_{i+1} makinesi boş olmadığı sürece robot tarafından taşınmaz. Makinelerdeki işlem zamanlarıyla ilgili 3 farklı durum söz konusu olabilir:

1. İşlemi biten parça istendiği zaman M_i makinesinden alınabilir. Bir sınırlama bulunmamaktadır (zaman sınırlamasız).
2. İşlemi biten parça işlem biter bitmez M_i makinesini terk edip, M_{i+1} 'e taşınmak zorundadır (beklemesiz).
3. İşlemi biten parçanın M_{i+1} makinesine taşınması için belirli bir zaman aralığı mevcuttur ve bu aralık tüm taşımalarda sabittir (zaman penceresi).



T:Toplamsal, Ö:Öklid, S:Sabit

Şekil 2.1: Robotik hücreler sınıflandırması [28]

Taşıma zamanlarının ölçümü için de farklı parametreler kullanılır. Bunlar, toplamsal (T), öklid (Ö) ve sabit (S) zaman ölçümleridir. Bunun dışında üretim stratejisinin seçimi de bu gruba girer. Aynı ya da farklı tip parçalar bu strateji parametreleridir. γ parametresi amaç fonksiyonunu temsil eder. Amaç fonksiyonu, birim çevrim süresini bulmaktır. Birim çevrim süresi 1-birim döngüleri olabileceği gibi k-birim döngüleri de olabilir. Bütün bu özelliklere göre Geismar vd. [28]'nin yapmış olduğu sınıflandırma Şekil 2.1' de görülmektedir.

Robotik hücrelerde çizelgeleme konusu aşağıdaki başlıklar altında incelenecektir.

2.1.1 Tek Tutuculu Robotik Hücrelerde Çizelgeleme

Tek tutuculu robotik hücreler için Brauner [13]' in 2008 yılındaki çalışması literatür taraması şeklindedir. Sethi vd. [72] m makineli sistem için $m!$ kadar 1-birim döngüsü bulunduğunu ve iki makineli sistem için 1-birimlik döngülerden birinin optimal olacağını ispatlamış, 3 makineli sistem için de optimal olabileceğini önermiştir.

Crama ve van de Klundert [19, 20] ise, 3 makine için bir birim döngülerin optimal olduğunu ispatlamışlardır ve m makineli robotik hücre için en iyi 1-birim döngüyü veren bir algoritma geliştirmişlerdir. Bu algoritma, mevcut çalışmada da önemli bir yer tutmaktadır. Bu yüzden ilerleyen bölümlerde daha ayrıntılı anlatılacaktır.

Brauner ve Finke [16, 15], 4 veya daha fazla makineli hücreler için 1-birim döngülerin optimal olamayabileceğini, k-birimlik döngülerin $k \geq 2$ de optimal sonuçlar verebileceğini göstermişlerdir.

Geismar vd. [30]'nin yapmış oldukları çalışmada makine sayısı ve işlem sıraları üzerinde değişiklikler yapılarak performansın nasıl etkilendiği incelenmiştir. İşlemlerin makinelerde gerçekleşme sırası değişmiş ve her deneme için makine sayısı artırılmıştır. 2 makineli robotik hücreler için verimlilikte bir artış yoktur. 3 ve 4 makineli robotik hücreler için verimlilik $\%14\frac{2}{7}$ artmıştır. Bu da işlemlerin makinelere atanmasının aynı tip parça üretimindeki etkisini göstermektedir.

Dawande vd. [22], sabit taşıma zamanlı robotik hücre çizelgeleme problemi üzerinde çalışmışlardır. Sabit taşıma zamanı, robotun herhangi iki makine

arasındaki taşıma zamanının sabit olduğunu ifade eder. Bu durum için optimal 1-birim döngüsünü bulan bir algoritma geliştirilmiştir.

Yapılan yeni çalışmalar daha çok üretimde esneklik üzerinedir. Literatürde üretimde esneklikle ilgili çok sayıda tanım yapılmıştır. Bu literatürde en çok ele alınanlar, aynı makinenin veya işçinin değişik işlemleri yapabilme kabiliyeti ve işlem sıralarının değiştirilebilme kabiliyeti şeklinde ifade edilen "makine" ve "süreç" esnekliği tanımlarıdır. Bu esneklikler sayesinde iş yükleri daha dengeli dağıtılabilir, işlerin tamamlanma zamanı azaltılarak toplamda üretilen miktar artırılabilir. Aktürk vd. [3], 2 makineli robotik hücreler için operasyon esnekliği üzerine çalışmışlardır. Operasyon esnekliği parçanın bazı operasyonlarının yer değiştirme veya değiş tokuş edilmesi gibi alternatif süreç planları ile üretilebileceği bir özelliktir [70]. Ayrıca operasyon esnekliği istasyon sayısından fazla iş olduğu ve işlerin tümünün yapılacak şekilde istasyonlara atanabileceği varsayımı altında, işlerin istasyonlara atanması ile ilgilidir [63]. Her parça bir takım operasyonlarla 2 makineli robotik hücrede üretilir ve her makine bu operasyonları gerçekleştirebilir. Problem parametrelerine bağlı olarak 1-birim veya 2-birim döngülerinin optimal olduğu parametre değerleri belirlenmiştir.

Daha sonra Gültekin vd. [33] 2 makineli robotik hücrelerde optimal döngüyü makinelerde kullanılan kesici u takımlarının makinelerdeki uç haznesi kapasitelerine bağlı olarak her makineye yerleştirilemediği durumları incelemişlerdir. Takım kısıtı eklendiğinde atanabilirlik matrisi de değişmektedir. Bazı işler birinci makinede gerçekleşirken, bu kısıtla birlikte bazı işleri ikinci makineye atamak gerekir. 1-birim ya da 2-birim döngüleri için optimallik bölgeleri belirlenmiştir.

Gültekin vd.[34]'nin 3 makineli robotik hücreler için de çalışması bulunmaktadır. Klasik robotik hücre çizelgelemesi literatüründe işlem zamanları bir parametre olarak alınmaktadır. Bu çalışmada ise işlem zamanları parametre değil, karar değişkeni olarak kabul edilmiştir ve işlemlerin makinelere atanmasına bağlı olarak değer almaktadır. İlk olarak esnekliğin getirdiği verimlilik kazancı araştırılmıştır. Daha sonra 1-birim ve 2-birim döngülerini için yeni problem doğrultusunda alt sınırlar belirlenmiştir. CNC makinelerin operasyonel esnekliği ve sürecin doğrudan bir sonucu olan yeni bir robot hareket döngüsü önerilmiştir. Önerilen yeni döngü 1-birim ve 2-birim döngülerinin tamamnında daha iyi sonuçlar vermektedir.

Gültekin vd. [35] daha sonra m makineli esnek robotik hücreler üzerinde çalışmışlardır. Önceki çalışmada önermiş oldukları döngünün 2-birimlik döngülerden daha iyi sonuçlar verdiğini kanıtlamışlardır. Ayrıca hat şeklinde olan robotik hücrelere göre robot merkezli üretim hücrelerinin kullanımının döngü zamanını azalttığını ispatlamışlardır. m makineli robotik hücreler için, önerilen döngülerin klasik robot döngülerini domine ettiği bölgeler belirlenmiştir. Ayrıca makine sayısını karar değişkeni olarak alıp sistemde kullanılacak olan optimal makine sayısını belirlemişlerdir.

Daha sonra Gültekin vd. [36] m makineli robotik hücreler için "sade döngü" (pure cycle) adını verdikleri döngüler üzerinde çalışmışlardır. Sade döngüler, esnekliğin bir sonucudur ve aynı makinelerden herhangi birinin, bir parçanın tüm işlemlerini yapabilecek kadar esnek olmasına dayanır. m makineli bir sistemde $m!$ kadar 1-birim döngüsü bulunur [72]. Fakat, sade döngüler söz konusu olduğunda m makineli bir sistem için $(2m - 2)!$ kadar döngü bulunmaktadır. 3 makineli bir robotik hücrede 6 tane 1-birim döngüsü varken 120 tane sade döngü bulunmaktadır. Sade döngüler için optimizasyon modeli belirlendikten sonra, 1-birim ve 2-birim döngüleri için optimal sonucu veren bölgeleri bulmuşlardır. Ayrıca, sade döngüler kümesinin esnek robotik hücreler hareket döngülerine göre daha domine olduğu kanıtlanmıştır.

Çok tip parça üretimi, farklı tip parçaların aynı hatta üretilmesidir. Farklı tip parçalar, üretimi gerçekleştirilecek herhangi bir makine için farklı üretim zamanlarına sahiptir. Küçük işletmelerin bu uygulamaları daha sık görülür. Hatta daha verimli kullanabilmek ve yeterli miktarda üretim yapmak için çok tip parça üretimini kullanırlar [28]. Çok tip parça üretimini esas alan çalışmalar da literatürde mevcuttur.

Hall vd. [39], 2 makineli hücreler için en iyi robot hareket döngüsünü ve parça sıralamasını $O(n^4)$ zamanda bulan bir algoritma geliştirmişlerdir. Srisankarajah vd. [74] verilen robot hareket döngüsü için en iyi robot hareket döngüsünü bulma problemini ele almış ve $2m-2$ kadar döngünün polinom zamanda çözülebileceğini ispatlamışlardır. Bunun için robot döngülerini 4 farklı durumda sınıflandırmışlardır. Bunlar sıralama probleminin olmadığı durum, polinom zamanda çözülebilen ve gezgin satıcı problemine benzeyen durum, gezgin satıcı problemine benzeyen fakat NP-zor çözümlü durum ve gezgin satıcı problemi yapısına benzemeyen

fakat NP-zor olan durumdur. Kamoun vd. [45] ise 3 ve 4 makineli hücrelerde parça sıralamasını bulmak için sezgisel bir algoritma geliştirmişlerdir. Batur vd. [6] 2 makineli esnek robotik hücreler üzerine çalışmışlardır. Gezgin satıcı modelinden yola çıkarak, ilk olarak 2 makineli robotik hücreler için matematiksel model oluşturmuşlar daha sonra 2-aşamalı sezgisel adını verdikleri etkili bir sezgisel geliştirmişlerdir. Geliştirdikleri sezgisel test edildiğinde ekstra yüklenme-boşaltma zamanları, bekleme zamanlarında artış ve makine kapasitesinin daha etkili kullanımı gibi durumlara rağmen sezgiselin literatürdeki sezgisellere kıyasla daha etkili olduğu görülmüştür. Kamalabadi vd. [44] çok tip parça üretiminde 3 makineli robotik hücreler için yeni bir matematiksel model geliştirmişlerdir. Geliştirilen model Petri ağlarına dayalıdır. Petri ağları 4 farklı bölümden oluşur: sonlu sayıda yer kümesi, sonlu sayıda değişim kümesi, sonlu sayılı bağlantılar ve ağırlık fonksiyonu. Her sonlu sayıda yer kümesinin bulunduğu bir nokta vardır. Bu noktalar değişim kümelerine atanırsa bu sisteme Petri ağları denir. Matematiksel modelle çözüm büyük ölçekli problemlerde NP-zor olduğundan, parça kümesi optimizasyonuna dayalı bir algoritma geliştirilmiştir. Geliştirilen algoritma küçük ölçekli problemlerde optimal sonucu bulurken, büyük ölçekli problemlerde de optimale yakın sonuçlar vermektedir. Fathian vd. [26] petri ağlarından yola çıkarak çok tip parça üretimi için 2 makineli robotik hücreler üzerine matematiksel model geliştirmişlerdir.

2.1.2 Çift Tutuculu Robotik Hücrelerde Çizelgeleme

Tek tutuculu sistemde m makine de $m!$ kadar 1-birim döngüsü olduğunu söylenmişti. Çift tutuculu robotik hücrelerde ise, sadece 2 makineli bir üretim hücresi için bile 52 farklı döngü ortaya çıkmaktadır [71]. Bu yüzden Sethi vd.[71] çalışmalarında çift tutuculu robotik hücrelerin tek tutuculu robotik hücrelere göre avantajlarını incelenmişler ve parametreleri verilen bir çift tutuculu robotik hücrenin verimliliğini hesaplayan bir algoritma geliştirmişlerdir.

Drobouchevitch vd. [24] m makineli çift tutuculu robotik hücre için 1-birim döngü sayısını tahmin eden bir algoritma geliştirmişlerdir. Ayrıca, bir tutucu ile diğer bir tutucunun birbirlerinin konumuna gelmesi ile pozisyon değiştirmesi durumu için optimal çözüm bulunmuştur. Ayrıca, tek ve çift tutuculu robotik hücreler için

verimlilik karşılaştırması yapmışlardır. Yapılan karşılaştırma sonrası çift tutuculu robotik hücrelerin tek tutuculu robotik hücrelere göre daha üretkendir.

Geismar vd.[29] çift tutuculu hücreler için temel kavramları ele almışlardır. Tek ve çift tutuculu hücreler için verimlilik kıyaslanmıştır. Tek ve çift tutuculu robotik hücreler için geliştirdikleri ayrı ayrı algoritmalarla tek tutuculu robotik hücrelerde kötü durum performansı %66,7, çift tutuculu robotik hücrelerde ise kötü durum performansı %77,8 olarak optimal çözüme yaklaşmıştır. Ayrıca çift tutuculu robotik hücreler için dominant 1-birim döngüsü bulunmuştur.

Çift tutuculu robotik hücrelerdeki 1-birim döngü sayısının tek tutuculu robotik hücrelerdeki 1-birim döngü sayısına göre çok daha fazla olduğunu daha önce belirtilmişti. Çok tip parça üretiminde ise problemin çözümü daha da zorlaşmaktadır. Sriskandarajah vd. [73] yaptıkları çalışmada 2 makineli sistem için sezgisel bir algoritma geliştirmişler ve daha sonra bu algoritmayı m makineli sisteme uyarlamışlardır. Geliştirdikleri sezgiselin hata oranı %10 'dan küçük çıkmıştır ve tek tutuculu robot yerine çift tutuculu robot kullanımının mevcut hücrede %18 ile %36 arasında bir iyileştirme sağladığını test etmişlerdir.

2.1.3 Çift Kollu Robotik Hücreler

Çift kollu robotik hücreler son yıllarda öne çıkan bir yapıdır. Bu yüzden çok fazla çalışma bulunmamaktadır. Bu konuda yarı iletken üretiminde kullanılan çift kollu küme araçları üzerine epeyce bir çalışma vardır. İlgili okuyucu için Lee [48]'nin çalışması bu alandaki çalışmaları derlemiştir. Bu konuda bildiğimize göre robotik hücre çizelgelemesi literatüründe yapılmış olan tek çalışma Geismar vd. [31]'nin çalışmasıdır. Geismar vd. [31] 2 ve 3 makineli hücreler için, iki kollu robotik hücrelerin tek ve çift tutuculu robotik hücrelere göre performanslarını değerlendiren bir çalışma yapmışlardır. Çift kollu robotik hücrelerin verimliliğinin tek tutuculu ve çift tutuculu robotik hücrelerin verimliliğine göre daha fazla olduğu kanıtlanmıştır.

2.2 Hat Dengeleme

Endüstride sıklıkla kullanılan montaj hatları da akış tipi üretim sistemlerinin bir türüdür. 1900lü yılların başında Henry Ford tarafından kullanılan ilk montaj hattından beri bu tip sistemler hala günümüzde önemini korumaktadır. Temposu değişmeyen bu hatlar, Henry Ford'un ilk yıllarından itibaren geliştirilerek, paralel istasyonlu hatlara, özelleştirilmiş karma ve karışık modelli hatlara, U tipi yerleşimli hatlara dönüşmüştür. Ancak bu dönüşüm, toplam iş yükünü hat boyunca üretim için montaj işleminin yapıldığı tüm istasyonlara dağıtılması sorunu olan montaj hatı dengeleme problemini de beraberinde getirmiştir [12].

Montaj hattı dengeleme 2 tip problem içerir [66]. Bunlardan birincisi, çevrim zamanının bir parametre olarak alınıp istasyon sayısının belirlenmesi problemidir. Diğeri ise, istasyon sayısının bir parametre olarak alınıp, çevrim zamanının bulunması problemidir. Çözüm yöntemleri olarak matematiksel ve sezgisel modeller kullanılmaktadır. Hat dengeleme sınıflandırmasında ilk olarak, klasik montaj hatları ile klasik montaj hatlarının geliştirilmesiyle ortaya çıkan montaj hatları arasında bir ayırım yapılmıştır [10]. Buna göre hatlar basit ve genel olmak üzere ikiye ayrılmış, daha sonra her bir dal çözüm yöntemlerine göre sınıflandırılmıştır. Montaj hattı dengeleme problemi üzerine geniş bir literatür mevcuttur. Scholl ve Becker [66]' in 2004 yılındaki literatür tarama makalesi, basit montaj hattı dengeleme problemi için literatür taraması şeklindedir. Genel montaj hattı dengeleme problemleri içinse, Becker ve Scholl [12]' un literatür tarama makalesi bulunmaktadır. Bu iki sınıf aşağıda daha detaylı incelenmiştir:

2.2.1 Basit montaj hattı dengeleme

Basit montaj hattı dengeleme adından da anlaşılacağı üzere, tek bir düz hatta makinelerin yerleşip işlemlerini gerçekleştirdiği problemidir. Montaj hattı dengeleme probleminin en yalın halidir ve tek modellenmiş deterministik görev süreleri mevcuttur. Problemin temel varsayımları şunlardır:

1. Montaj hattında tek çeşit ürün üretilir.
2. İşlemlerin makinelerde işlenmesine bağlı işlem süreleri deterministiktir.

3. İşlemler arası öncelik ilişkileri mevcuttur. Bir işlemin öncülleri, o işlemin atanacağı istasyondan daha arkada olan bir istasyona atanamaz.
4. Her işlem bir istasyona atanmalıdır fakat bir istasyona birden fazla işlem atanabilir.

Basit montaj hattı dengeleme problemi için matematiksel model kullanılarak pek çok çözüm yöntemi ele alınmıştır. Dar-El [21] 1975 yılında yaptığı çalışmada basit hat dengeleme problemi için sezgisel bir algoritma ve bu algoritmanın verimliliğini değerlendirmek için test verileri geliştirmiştir. Bu çalışmasında kullandığı test verilerinden bir tanesi incelenilen problemin test aşamasında da kullanılmıştır. Scholl vd. [68], daha önce geliştirilen bir algoritmaya(SALOME [69]) atama kısıtı ekleyerek yeni bir algoritma (ABSALOM) geliştirmişlerdir. Bu çalışma için kullanılan veri setleri, incelenilen problemin test aşamasında da kullanılacaktır. Pastor ve Ferrer [59] ise işlemlerin makinelere atanmasıyla ortaya çıkan atanabilirlik sayısının makine sayısına ve çevrim zamanına bağlı olmamasını sağlayan yeni kısıtları günümüze kadar kullanılan modellere tanıtmışlardır. Peeters ve Degraeve [60], basit hat dengeleme için temel kısıtlar kullanılarak oluşturulan matematiksel modelden yola çıkarak ve bazı doğrusal programlama teknikleriyle yeni bir alt limit oluşturmuşlardır. Gürsoy ve Nuriyev [37] de basit montaj hattı dengeleme için yalnız üretim teknikleri kullanarak matematiksel model geliştirmişlerdir. Yalnız üretim, stoklama ihtiyacını ortadan kaldırmayı ve düşük maliyetli, yüksek kaliteli üretimi hedefler. Ritt ve Costa [62], günümüze kadar kullanılan basit hat dengeleme problemine ait genel matematiksel modelleri inceleyip çözüm aralığını daraltan kısıtlar bularak daha etkin sonuçlar elde etmişlerdir. Matematiksel model için geliştirdikleri öncelik kısıtı ifade şekli, bu çalışma için oluşturulan matematiksel modelde de kullanılmıştır.

Basit montaj hattı dengeleme problemi için çözüm yöntemi geliştirilirken sezgisel metodlar da kullanılmıştır. Bautista ve Pereira [7], Sınırlı Dinamik Algoritma adını verdikleri sezgisel bir algoritma geliştirmişlerdir. Basit hat dengeleme problemi için geliştirilen bu algoritma literatürde yer alan diğer algoritmalar ile kıyaslandığından en iyi sonuçları vermiştir. Park vd. [57], basit hat dengeleme problemi unusrularından yola çıkarak elektrikli ev aletleri üzerine üretim yapan bir şirkette ortaya çıkan bir problem üzerinde çalışmışlardır. İşlemler 2 farklı özelliكتedir. Bazı işlemler basit hat dengeleme problemleri özelliklerini taşıırken, bazıları ise aynı

istasyona atanamaz ve atanabilmesi için basit hat dengeleme problemi özellikleri taşıyan işlemlerin atamalarının yapılıp, bu işlemlerin atanamadığı ara makinelere atanması gerekir. Bu probleme göre öncelik diyagramlarının görevler arasındaki öncelikleri tanımlamaya yeterli bulunmamaktadır. Geliştirdikleri sezgisel adımları ağ teoremi ile benzer özellikler taşır. Ağ teoremi, nesnel arasında simetrik ve asimetrik ilişkileri gösteren diyagramdır. Sezgisel ile test aşamasında optimal sonuca yakın sonuçlar elde edilmiştir.

Bilinen en eski yöntemlerden biri Pozisyon Ağırlığı Yöntemi (Ranked Positional Weight)'dir [40]. Bu yöntemde bir görevin kendi süresi ile ardılı olan görev süreleri toplamı o görevin pozisyon ağırlığını verir. Görevler pozisyon ağırlıklarına göre sıralanır. En yüksek pozisyon ağırlığına göre görevlerin istasyonlara ataması yapılır. Tüm görevler istasyonların çevrim zamanı aşılmayacak şekilde atanır.

Hoffman algoritması [41], öncelik ilişkileri matrisindeki sütündeki tüm değerler toplanarak, her birine kod numaraları verilir. İlk dize seçilir ve bu dizeden ilk 0 olan eleman seçilir. Dizedeki bu eleman o işlemin önceliğinin olmadığını göstergesidir. Bu işlem ilk istasyona atandıktan sonra matris yeniden oluşturulur. Daha sonra Fleszar ve Hindi [27], Hoffman sezgiseline[41] bağlı kalarak farklı bir sezgisel geliştirmişlerdir. Bu sezgisel de öncelik ilişkilerine dayanarak en iyi sonucu seçer, öncelik ilişkilerine Hoffman [41] sezgiselindeki gibi tek yönlü olarak bakmaz, matrisi 2 taraflı inceler ve Hoffman sezgiseline [41] göre daha iyi sonuçlar verir.

COMSOAL methodu [5], öncelik diyagramına göre oluşturulan atanabilir işler arasından rastgele seçim yaparak görevleri istasyonlara sırasıyla atar, bu işlemi defalarca tekrarlar. Tekrarlar arasından en iyi çözümü esas alır.

Hackman [38]'in geliştirdiği Immediate Update First Fit (IUFF) yöntemi 6 fonksiyona dayalı bir yöntemdir. Bu fonksiyonlar, pozisyon ağırlığı, ters pozisyon ağırlığı, ardıl sayısı, komşu ardıl sayısı, öncül sayısı ve görev süresidir. Bu yöntem atanabilecek görevlere bu fonksiyonlara göre değer belirleyerek öncelik tanımlar.

Baybars [11]'in geliştirdiği Rank and Assign yöntemi de IUFF yöntemi ile benzerlik gösterir. 6 fonksiyondan biri seçilerek, tüm görevler fonksiyon değerlerine göre büyükten küçüğe doğru sıralanır ve bu sıralama üzerinden atama yapılır.

Basit montaj hattı dengeleme için, yukarıda bahsedilenlerin dışında birçok sezgisel yöntem bulunmaktadır. Literatürde karşılaşılan diğer sezgisel yöntemler şunlardır:

- Aşamalı sıralama (Jackson [42])
- Aday matris (Salveson [65])
- İlişkili etkinlik (Agrawal [2])
- İki aşamalı dengeleme (Moddie-Young [53])
- Killbridge-Wester yöntemi[46]
- Probabilistik hat dengeleme (Elsayed-Boucher [25])
- Raouf-Tsui-Elsayed yöntemi[61]
- Gruplama yöntemi (Tonge [75])

Literatürde özellikle Jackson [42], Killbridge-Wester [46] ve Salveson [65]' un geliştirdiği yöntemler ön plana çıkmaktadır.

2.2.2 Genel Montaj Hattı Dengeleme

Genel montaj hattı dengeleme problemi gerçek hayatta karşılaşılan problemleri ele almak üzere paralel çalışan istasyonlar ya da işlemler, müşteri odaklı modeller, U tipi yerleşimli hatlar gibi alternatif durumların ele alındığı problemdir. Tek modellenen deterministik montaj hattı dengeleme problemlerinin dışındaki tüm problemleri içerir. Yani basit montaj hattı dengeleme probleminin karakteristiklerinin bir veya daha fazlasının gevşetilmiş halidir [10].

Literatürde genel montaj hattı dengeleme problemi için çözüm yöntemi olarak matematiksel model kullanılan pek çok araştırma mevcuttur.

Ağpak ve Gökçen[1] en küçük istasyon sayısını bulmak için 0-1 tamsayılı model geliştirmişlerdir. Kaynak kısıtlı hat dengeleme problemi (Resource Constrained Assembly Line Balancing, RCALBP) üzerine çalışmışlardır. Bu tip problemde sınırlı sayıda işçi ve sınırlı sayıda makine bulunmaktadır. Yapılan çalışmaya ek

olarak gelecek çalışmalarda, kullanılan bir kaynak için istasyon sayısı belirlenebilir ya da en küçük döngü zamanı bulunabilir.

Gökçen ve Erel [32], çok sayıda parçanın tek bir hatta üretildiği karışık modelli hat dengeleme problemi üzerine çalışmışlardır. Bu problem türü, müşteri istekleri doğrultusunda ortaya çıkmıştır. Çeşitli tipte ürün elde edebilmek için büyük stoklara gerek kalmadan tek bir hatta ürünler üretilir. Bu problem üzerine tamsayı matematiksel model geliştirilmiştir.

Bağımlı sıralı hat dengeleme problemini (Sequence Dependent Assembly Line Balancing Problem, SDALBP) basit montaj hattı dengeleme probleminden ayıran en önemli özellik, öncelik diyagramına alternatif arkların da eklenmesidir. Scholl vd. [67] bu tip problemler üzerine tam sayılı matematiksel model geliştirmiştir.

Hat dengeleme tasarım problemi de makinelerin yerleşimi üzerine kurulu olan bir problemdir. Atanacak işler, birbirinin aynısı olmayan makinelerde üretilebilmektedir. Makine çeşitliliği de bir kısıt olarak modele dahil olur. Nicosia vd. [54] bu problem üzerine dinamik programlama algoritması geliştirmiştir.

Genel montaj hattı dengeleme probleminin çözüm yöntemleri göz önüne alındığında sezgisel yöntemler en sık karşılaşılanıdır. Robotik hat dengeleme probleminde (Robotic Assembly Line Balancing Problem, RALBP), farklı robotlar farklı görevlere atanmıştır ve her robot kapasite ve özelliklerinden dolayı farklı montaj zamanlarına sahiptir. Bu çalışmada ele alınan robotik hücreler ile RALBP problemi kapsamında ele alınan problemler farklılık göstermektedir. Temel fark, bu çalışmada ele alınan hücrelerde robotlar malzeme taşımaktan sorumludur ve üretim hücresinde bir tane robot bulunmaktadır. RALBP problemlerinde ise robotlar parçalar üzerinde işlem yapmaktadırlar ve sistemde birden fazla robot bulunmaktadır. Levitin vd. [49], bu tip problem üzerine genetik algoritma geliştirmişlerdir.

Miralles vd. [52], montaj hattı işçisi atama ve dengeleme problemi (Assembly Line Worker Assignment and Balancing Problem, ALWABP) üzerine dal sınır algoritması ve sezgisel bir algoritma geliştirmişlerdir.

Alternatif alt gruplar hat dengeleme probleminde (Alternative Subgraphs Assembly Line Balancing Problem, ASALBP), her grup üretim ya da üretim

süreçlerinin farklı işlemlerini içerir. Her bir alt grup geliştirilebilir görev ve işlem sürelerini tanımlayan bir öncelik alt grafiğini temsil eder. Capacho vd. [17], öncelik ilişkileri ve işlem zamanları gibi değerleri değiştirip, bunlara uygun sezgiseller geliştirip değerlendirmişlerdir.

İki taraflı hatlar (Two-sided Assembly Line Balancing Problem, TALBP), sağ ve sol hattın paralel olduğu hatlardır. Özcan ve Toklu [56], bu tip problemler için bir tabu arama geliştirmişlerdir. Kim vd. [47] de bu tip problemler üzerine çalışmış, matematiksel modelin yanısıra genetik algoritma geliştirmişlerdir. Xiaofeng vd. [78], Hoffman [41] algoritmasından yola çıkarak bu özel problem için bir sezgisel algoritma geliştirmişlerdir. Özcan ve Toklu [56]'nın yapmış olduğu çalışmada bazı problemler için optimal çözümler bulunmuştur.

Makine hazırlık sürelerinin genel montaj hattı dengeleme problemine dahil olmasıyla ortaya farklı bir problem çıkmaktadır (General Assembly Line Balancing Problem With Setups, GALBPS). Martino ve Pastor [51] bunun için öncelik ilişkilerini de dikkate alarak, bir sezgisel model geliştirmişlerdir.

Malakooti ve Kumar [50], çok kriterli hat dengeleme problemi için bilgi tabanlı sistemleri kullanmışlardır (Multi Objective Assembly Line Balancing Problem, MOALBP). Çok yönlü hat dengeleme probleminde farklı amaç fonksiyonları bir arada ele alınmaktadır. İstasyon sayısını enküçükmek, çevrim zamanını enküçükmek gibi amaç fonksiyonlarının dışında stokları enküçükmek ya da maliyeti enküçükmek gibi amaç fonksiyonları da kullanılabilir. Çok kriterli bu hatlar uzman sistemlerle ilişkilendirilir. Kendi sistemlerini ve veri tabanlarını oluşturarak en uygun sezgiseli bulmuşlardır.

Jin ve Wu [43], daha önce bahsedilen karışık modellenmiş montaj hattı dengeleme problemi üzerine çalışmışlardır. Bu hatlarda makine çeşitliliği söz konusudur. Daha önce yapılan çalışmalarda zamanında üretim için hedef arayan algoritmalar kullanılmıştır. Bu aç gözlü (greedy) algoritmalar, anlaşılması kolay olduğundan tercih sebebidir. Daha sonra bu algoritmalarla yola çıkarak, varyans algoritması adını verdikleri bir sezgisel geliştirmişlerdir ve bu sezgiselin diğerlerine göre daha iyi sonuçlar verdiği gösterilmiştir.

Boysen ve Fliedner [14], U tipi yerleşik hat dengeleme problemi için versatil algoritma adını verdikleri bir sezgisel geliştirmişlerdir. U tipi yerleşik hatlar, hattın

giriş ve çıkış istasyonunun aynı yönde olduğu ve işçinin hattın her iki tarafında görevlerini yerine getirebileceği hatlardır.

Farklı tip hatlara yönelik gerçekleştirilen farklı tip sezgisellerin yanında hat dengeleme problemleri için ayrıca karınca kolonisi algoritmaları ile de çalışılmıştır [9, 8, 64, 76]. Bunun yanında tabu arama algoritmaları da en sık karşılaşılan çözüm yöntemlerinden biridir [55, 4, 23, 58]. Literatürde, bunun dışında pek çok hat dengeleme problemi ve sezgisel çeşidi bulunmaktadır. Bu yöntemler Scholl ve Becker [66] ve Becker ve Scholl [12]' un tarama makalelerinde incelenmiştir.

2.3 Özet

Literatür araştırması bölümünde robotik hücrelerde çizelgeleme ve montaj hattı dengeleme konuları ve her iki konunun alt başlıkları için yapılmış çalışmalar incelenmiştir.

Bu tez çalışmasında ele alınan problem özellikle üretimde esneklik kapsamında robotik hücrelerde yapılan çalışmalar ile problem tanımı açısından benzerlik göstermektedir. Crama ve van de Klundert [19, 20]' in m makineli robotik hücreler için oluşturdukları optimal döngüyü veren algoritmalarından da faydalanılarak, m makineli robotik hücreler için bir sezgisel geliştirilmiştir.

Problem için diğer bir unsur da hat dengelemedir. Bir robotik hücrede bulunan makinelere uygun iş dağılımının gerçekleşmesini sağlamak, tez çalışmasındaki problemde de mevcuttur. Hat dengeleme problemlerine ait literatür incelendiğinde daha önce robotik hücrelerle ilgili çalışma yapılmadığı gözlemlenmiştir. Genel montaj hattı dengeleme problemleri arasında robotik hat dengeleme problemi ile bu tez çalışmasında ele alınan problemler karıştırılmamalıdır. Robotik hat dengeleme probleminde robotlar makineler yerine kullanılırken, bu problemde robotlar malzeme taşıma sistemleri olarak kullanılmaktadır. Basit hat dengeleme probleminde ise makinelere dengeli bir şekilde işlemlerin atanması gerçekleşmektedir fakat bu tip çalışmalarda da robotlar malzeme taşıma da kullanılmamıştır.

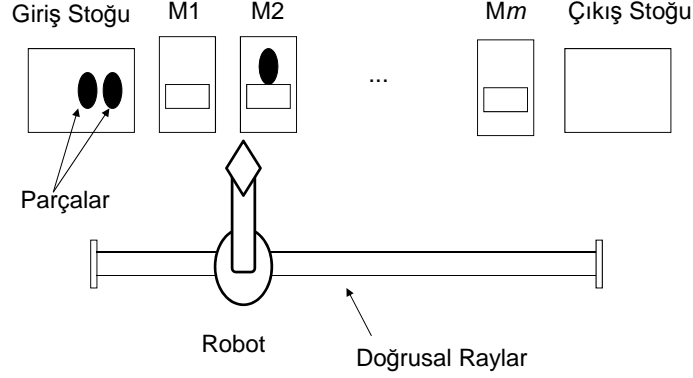
Literatür genel olarak incelendiğinde, problem üretimde esneklik kapsamında robotik hücrelerde yapılan çalışmalar ve robotik hat dengeleme problemine benzese bile, daha önce bu şekilde bir problem tanımı ile çalışılmamıştır. Bu çalışmada ele alınan sistem yeni bir robotik üretim hattı kurulacağı zaman çözülmesi gereken önemli bir tasarım problemi olabileceği gibi, aynı zamanda robotik hücrelerin montaj hattı dengeleme sistemleri konusunda da yer alabileceğini gösterir. Bu iki farklı konunun entegre edildiğinde sistemde düzenli bir malzeme akışı sağlanır, insan gücü ve tezgah kapasiteleri kullanımı artar. m makineli robotik hücreler için yapılan bu çalışma, en genel problemin çözümü ve zorluğu için öngörüler elde etmek adına önem taşımaktadır.

3. Problem Tanımı ve Eniyileme Modeli

Bu bölümde, çalışmaya konu olan problem detaylı bir şekilde tanımlanmıştır ve problemin zorluğu ile ilgili bilgiler verilmektedir. Problemi anlamak için gerekli olan temel kavramlar ve notasyona yer verilmekte, ayrıca problemi kesin bir şekilde çözmek için geliştirilen matematiksel modeller de bu bölümde sunulmaktadır.

3.1 Temel Kavramlar ve Problem Tanımı

Bu çalışmada sürekli aynı tip parçanın üretildiği ve m makineden oluşan bir montaj hattı ele alınmaktadır. Klasik montaj hatlarından farkı makineler arası parça transferi ve makine yükleme boşaltma işlemlerinin Şekil 3.1'de görüldüğü gibi bir robot tarafından yapılmasıdır. Robot doğrusal raylar üzerinde hareket etmektedir ve parçalar üzerinde herhangi bir işlem yapmamaktadır. İşlenecek her parçanın makinelerde işlenmek üzere toplam N tane değişik işlemleri bulunmaktadır. j işleminin işlem süresi o_j $j = 1, \dots, N$ ile gösterilmektedir. Sistemdeki makinelerin değişik işlemleri yapabildiği varsayılmaktadır. Bu durum gerçek hayatta bir çok endüstride karşılaşılan bir durumdur. Örneğin, sistemde CNC makinelerin kullanılması durumunda bu varsayım geçerli olmaktadır. Çünkü bu makineler, kesici uç haznelerinde gerekli kesici uç takımları bulunması halinde değişik işlemleri yapabilmektedirler. Bunun bir sonucu olarak bu çalışmada bir robotik üretim hattı tasarımı konusu ele alınmaktadır. Hangi işlemin o işlemi yapabilecek kabiliyette olan makinelerden hangisi tarafından yapılacağı bir karar değişkenidir.



Şekil 3.1: m makineli robotik hücre

İşlemlerin makinelere atanabilirlikleri a_{ij} parametresi ile gösterilmektedir. $a_{ij} = 1$ ise j işlemi i makinesinde işlenebilir, $a_{ij} = 0$ ise de işlenemez anlamına gelmektedir. ($i = 1, \dots, N, j = 1, \dots, m$)

İşlemlerin makinelere atanmasında etkili olan bir diğer faktör de işlemler arasındaki öncelik ilişkileridir. Klasik montaj hattı dengeleme problemlerinde de olduğu üzere bazı işlemlerin yapılabilmesi için diğer bazı işlemlerin tamamlanmış olması gerekir. Bu durum da b_{kj} parametresi ile gösterilmektedir. Eğer k işi j işinin hemen öncesi ise $b_{kj} = 1$ değilse $b_{kj} = 0$ değerini almaktadır. ($k, j = 1, \dots, N$)

İşlemlerin makinelere atanması belirlendikten sonra, parçaların makinelerdeki toplam işlem zamanları kolaylıkla belirlenebilir. Bu atamalar aşağıdaki değişkenle gösterilecektir;

$$x_{ij} = \begin{cases} 1; & \text{Eğer } j \text{ işi } i \text{ makinesine atanmışsa} \\ 0; & \text{Değilse} \end{cases}$$

Bu değişken tanımı kullanıldığında i makinesindeki toplam işlem süresi aşağıdaki gibi hesaplanır.

$$p_i = \sum_{j=1}^N x_{ij} o_j \quad (3.1)$$

Ayar süreleri işlem sürelerine dahil olup önceden bilindiği varsayılmaktadır. Problem parametreleri deterministik özellikler taşır, önceden belirli ve sabittir. Ayrıca bir işe başlandığında o iş kesilmeden tamamlanana kadar devam etmektedir. Atamalar haricinde, problemde karar verilmesi gereken bir diğer önemli unsur

robotun makineleri hangi sıra ile yükleyip boşaltacağı ve parça transferlerini nasıl bir sıra ile yapacağıdır. Robotik hücre çizelgeleme literatüründe yaygın olduğu üzere, robotun makineleri yükleme ve boşaltma zamanlarının he biri ϵ , ardışık iki makine arası hareket zamanı ise δ olarak gösterilecektir. Robot hareket zamanlarının toplanabilir olduğu varsayılmaktadır. Yani, robotun i makinesinden l makinesine toplam gidiş zamanı $|i - l|\delta$ kadardır. Sistemdeki giriş stoğu 0, çıkış stoğu ise $m + 1$ numaralı makine gibi düşünülmektedir.

Bu çalışmada, Davande vd. [22]'nin optimal çözümler verdiğini ispatladığı "döngüsel robot hareket sıraları" ele alınacaktır. Buna göre robot belirli bir sistem durumundan başlar, bazı aktiviteleri gerçekleştirir ve başlangıç durumuna geri döner. Bu hareket dizisi sürekli tekrarlanır. Burada "aktivite (A_i)" olarak ifade edilen kavram Crama ve van de Klundert [19] tarafından aşağıdaki şekilde tanımlanmıştır:

1. robotun (M_i)' den parça alması,
2. parçayı (M_{i+1})'e taşıması,
3. parçayı (M_{i+1})'e yüklemesi.

Bütün robot döngüleri yukarıda belirtilen aktivitelerle tanımlanabilir. Hattın ardı ardına iki ürün çıkarması arasında geçen süre yani, bir parçanın üretimi için gereken ortalama zamana "döngü zamanı" denir. 1-birim döngülerinde her bir aktivite bir defa tekrar edilir. Bu çalışma kapsamında uygulaması kolay, pratik, anlaşılması kolay ve literatürde çok iyi sonuçlar verdiği ispatlanmış olan 1-birim döngüler ele alınacaktır. Daha önce bahsedildiği üzere m makineli bir sistemde $m!$ adet 1-birim döngüsü bulunmaktadır [72]. Buna göre, 3 makine için 6, 4 makine için 24 ve 5 makine için 120 1-birim döngüsü bulunmaktadır. Dolayısıyla, en iyi 1-birim döngüsünün bulunması hala karmaşık bir problemdir. Yapılmış olan aktivite tanımları ile robot hareket döngülerinin nasıl ifade edilebileceğini göstermek için 3 makineli bir sistemi ele alalım. Genelliği kaybetmeden bütün döngülerin robotun giriş istasyonundan bir parça almaya başlamasıyla başladığını varsayabiliriz. Buna göre 3 makine için S1-S6 ile gösterilen aşağıdaki 6 robot hareket döngüsü bulunmaktadır.

S1: $A_0A_1A_2A_3$

S2: $A_0A_1A_3A_2$

S3: $A_0A_2A_1A_3$

S4: $A_0A_2A_3A_1$

S5: $A_0A_3A_1A_2$

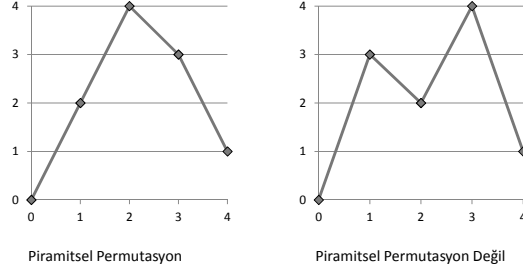
S6: $A_0A_3A_2A_1$

Örneğin, S3 döngüsünde robot giriş stoğundan bir parça alarak başlar, sırasıyla ilk makineyi yükler, M_2 'deki parçayı M_3 'e taşır, M_1 'deki parçayı M_2 'ye taşır. M_3 'teki bitmiş parçayı çıkış stoğuna bırakır ve giriş stoğuna geri döner.

Bu tanımlamalardan sonra, bu çalışmada ele alınan problem, üretim hızını en büyükecek şekilde işlemlerin öncelik ve atanabilirlik ilişkilerine göre makinelere atanması ve en iyi 1-birim robot döngüsünün bulunmasıdır.

3.2 Kesin Çözüm Yöntemi

Bu kısımda problemin kesin çözüm yöntemine ulaşmak için matematiksel modelleme tabanlı bir kesin çözüm yöntemi geliştirilecektir. Bir önceki bölümde tanımlandığı üzere problem iki kısımdan oluşmaktadır. Bunlardan ilki işlemlerin makinelere atanmasını bulmak, diğeri ise en iyi 1-birim döngüsünü bulmaktır. Bu iki problem birbirinden bağımsız değildir, çünkü en iyi 1-birim döngü makinelerdeki işlem sürelerine bağlıdır. Makinelerdeki işlem sürelerini ise işlemlerin makinelere atanması belirlemektedir. Dolayısıyla bu iki problem aynı anda çözülmelidir. Fakat en iyi 1-birim döngüsünü bulmak için herhangi bir matematiksel model kurmak mümkün değildir. Bu sebeple, her robot hareket döngüsü tek tek ele alınmalı, her birisi için en iyi işlem atamaları bulunmalı ve sonrasında, bütün döngüler birbirleriyle karşılaştırılarak en iyi çözüm bulunmalıdır. Daha önce bahsedildiği üzere m makineli bir sistemde $m!$ tane 1-birim döngüsü bulunmaktadır [72]. Dolayısıyla, $m!$ tane ayrı model kurulmalı ve çözdürülmelidir. Bu noktada, Crama ve van de Klundert [19]'in ispatladıkları bir sonuç çözdürülecek model sayısını önemli ölçüde azaltmaktadır. Buna göre, öncelikle "piramitsel permutasyon" şeklinde uygun döngüler ayrıştırılmıştır. Piramitsel



Şekil 3.2: Piramitsel permutasyonun örnek gösterimi: $A_0A_2A_4A_3A_1$ döngüsü ve Piramitsel permutasyona örnek olmayan gösterim: $A_0A_3A_2A_4A_1$ döngüsü

permutasyonlar, A_0 aktivitesiyle başlamak üzere, aktivite dizisindeki önce arttığı, bir tepe noktasına ulaştıktan sonra ise azaldığı tek kubbeli indis sıralamalarıdır. Şekil 3.2' de piramitsel permutasyon olan $A_0A_2A_4A_3A_1$ ve piramitsel permutasyon olmayan $A_0A_3A_2A_4A_1$ döngülerinin indis yapıları görülmektedir.

Piramitsel permutasyonların önemi, Crama ve van de Klundert [19]'in optimal döngülerin sadece piramitsel permutasyonlar arasında olabileceğini ispatlamasından sonra artmıştır. Bu ispat, $m!$ olan alternatif döngü sayısını 2^{m-1} 'e indirmektedir. Örneğin 6 makineli bir sistemde toplam 1-birim döngü sayısı $6! = 720$ iken, piramitsel döngü sayısı $2^5 = 32$ ' dir. Her ne kadar çözdürülmesi gereken matematiksel model sayısı önemli ölçüde azalmış olsa da, toplam sayı hala makine sayısına bağlı olarak üstel artış gösteren bir değerdir. Bu sebeple, matematiksel model ile ancak makine sayısının düşük olduğu durumlar için çözüm bulunabilir. Daha çok makinenin yer aldığı sistemler için ise başka bir çözüm yöntemi geliştirilmesi gerekmektedir. Bu amaçla geliştirilmiş olan sezgisel Bölüm 4'te anlatılacaktır. Fakat, hem düşük makine sayılarında kullanılmak üzere, hem de sezgiselin performansını test etmek üzere önce matematiksel model yöntemi anlatılacaktır.

Matematiksel modelin kullanılabilmesi için öncelikle her alternatif döngünün çevrim zamanı ifadesi hesaplanmalıdır. Bunun nasıl yapıldığını 4 makineli bir sistemdeki $A_0A_1A_3A_4A_2$ döngüsünü kullanarak açıklayalım: Bu döngüde robot önce giriş istasyonundan bir parça alır (ϵ), M_1 makinesine gider (δ) ve bu makineyi yükler (ϵ), bu makinede parça işlenirken robot makinenin önünde bekler (p_1) işlemi biten parça M_1 'den boşaltılır (ϵ). Daha sonra robot M_2 makinesine gider (δ) ve parçayı bu makineye yükler (ϵ). Parça makinede işlenirken robot önünde beklemeyi ve M_3 makinesine hareket eder (δ) ve dolu olan M_3

makinesinde gerekiyorsa parçanın işlenmesini bekler (w_3). İşlemi biten parçayı M_3 makinesinden alır (ϵ). Robot parçayı M_4 makinesine taşır (δ). M_4 makinesine parçayı yükler (ϵ) ve makinenin önünde parçanın işlenmesini bekler (p_4). İşlemi biten bu makineyi boşaltır (ϵ) ve çıkış istasyonuna hareket eder (δ), parçayı bu istasyona bırakır (ϵ). Daha sonra M_2 makinesine geri döner (3δ) ve burada işlenen parçanın işleminin bitmesini bekler (w_2 :). M_2 makinesini boşaltır (δ) ve M_3 makinesine gider (ϵ). M_3 makinesine parçayı yükler (ϵ). Daha sonra giriş istasyonuna geri döner (3δ).

Döngü zamanı yukarıdaki döngüde gerçekleştirilen tüm işlerin toplamıdır. Robotun makine önünde bekleme zamanları ise, iki şekilde gerçekleşebilir. Bunlardan ilki tam beklemedir ve robot makineyi yükledikten sonra o makinenin önünde işlemin bitmesini bekler. Tam bekleme zamanı, parçanın makinedeki işlem süresine eşittir. Diğer bekleme alternatifi ise kısmi beklemedir. Bu tip beklemede robot bir makineyi yükledikten sonra önünde beklemez ve farklı aktiviteler yapar. Makineyi boşaltmak için geri döndüğünde ise parçanın işlemi bitmişse bekleme yapmadan boşaltır, değilse kalan işlem süresi kadar bekler. Sonuç olarak elde edilen döngü zamanı aşağıdaki gibidir:

$$T: 10\epsilon + 12\delta + p_1 + p_4 + w_2 + w_3$$

$$w_2 : \max\{0, p_2 - 4\epsilon + 6\delta - p_4 - w_3\}$$

$$w_3 : \max\{0, p_3 - 4\epsilon + 6\delta - p_1\}$$

4, 5 ve 6 makineli robotik hücreler için bütün piramitsel permutasyonlu döngüler ve bunların çevrim zamanları Ek.1 'de görülebilir. Bu döngülerin her birisi için, çevrim zamanını enküçükleyecek şekilde, öncelik ve atanabilirlik kısıtlarını da dikkate alarak, işlemlerin makinelere atamasını belirleyen bir matematiksel model geliştirilmelidir. $A_0A_1A_3A_4A_2$ döngüsü için geliştirilmiş olan aşağıda matematiksel model örnek olarak verilmektedir. Modelde kullanılan parametreler aşağıdaki gibidir:

o_j : j işleminin süresi

ϵ : Robot yükleme-boşaltma zamanı

δ : Ardışık 2 makine arası robot hareket zamanı

$$a_{ij} = \begin{cases} 1; & \text{Eğer } j \text{ işi } i \text{ makinesine atanabilirse} \\ 0; & \text{Değilse} \end{cases}$$

$$b_{kj} = \begin{cases} 1; & \text{Eğer } k \text{ işi } i \text{ işinin hemen önceliği ise} \\ 0; & \text{Değilse} \end{cases}$$

Modelde kullanılan değişkenler ise şu şekildedir:

$$x_{ij} = \begin{cases} 1; & \text{Eğer } j \text{ işi } i \text{ makinesine atanmışsa} \\ 0; & \text{Değilse} \end{cases}$$

w_i : Parçayı boşaltmadan önce robotun i makinesindeki bekleme zamanı

Bütün döngüler için kurulmuş olan optimizasyon modelleri incelendiğinde amaç fonksiyonunun döngü zamanına bağlı olarak değiştiği ve bekleme zamanlarının döngü zamanlarına göre değişiklik gösterdiği söylenebilir.

$$\text{Min} \quad 10\epsilon + 12\delta + \sum_{j=1}^N x_{1j}o_j + \sum_{j=1}^N x_{4j}o_j + w_2 + w_3 \quad (3.2)$$

$$\text{s.t.} \quad w_2 \geq \sum_{j=1}^N x_{1j}o_j - 4\epsilon - 6\delta - \sum_{j=1}^N x_{4j}o_j - w_3 \quad (3.3)$$

$$w_3 \geq \sum_{j=1}^N x_{3j}o_j - 4\epsilon - 6\delta - \sum_{j=1}^N x_{1j}o_j \quad (3.4)$$

$$x_{ij} \leq a_{ij} \quad \forall i, j \quad (3.5)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad \forall j \quad (3.6)$$

$$\sum_{h \leq i} x_{hk}b_{kj} \leq \sum_{h \leq i} x_{hj} \quad \forall k, j = 1, \dots, N \quad (3.7)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \quad (3.8)$$

$$w_2, w_3 \geq 0 \quad (3.9)$$

Bu modelde amaç fonksiyonu döngü zamanını minimize etmektedir. Bekleme zamanlarına ait (3.3) ve (3.4) numaralı kısıtlar, farklı döngülere göre farklılıklar

gösterebilir. Bekleme zamanları döngü zamanı hesaplanırken ortaya çıkar. Bazı döngülerde bekleme zamanı yoktur ya da bekleme zamanının yukarıda hesaplanan değerden büyük olmaması gerekir.

(3.5) numaralı kısıt atanabilirlik kısıtıdır. Her işin atanabileceği makinelerden birisine atanmasını sağlar. İşlemler ve makineler esas alındığında, bir parçanın tamamlanabilmesi için her işlemin gerçekleştirilmesi gerekir. Bu kısıt modelde (3.6) numaralı eşitlikle gösterilmiştir.

Öncelik ilişkileri atamalarda göz önüne alınmalıdır. Bu kısıt daha önce Ritt ve Costa [62]'nin alternatif yazım şekilleriyle karşılaştırdıktan sonra en iyi performansı verdiği gösterilmiş olan kısıttır. Buna göre, (3.7) numaralı kısıt bunu sağlamaktadır. Eğer k işi, j işinin hemen öncesi ise k işi j işinin atandığı makineden daha önde olan bir makineye atanmalıdır. (3.8) ve (3.9) numaralı kısıtlar işaret kısıtlarıdır.

Oluşturulan en iyileme modeli, Intel Xeon *E5645 2x2.4 Ghz* işlemcili, 12 GB belleğe sahip bir makinede 8 paralel iş parçası (thread) olarak GAMS 12.2 kullanılarak çözdürülmüştür. Denemelerde makine sayısı veya işlem sayısı arttıkça çözüm zamanının çok hızlı arttığı gözlemlenmiştir. Döngüler ayrı ayrı incelendiğinde, bazı döngüler için modelin çözümü oldukça basittir. 4 makineli robotik hücre için, $A_0A_1A_2A_3A_4$ döngüsünü ele alalım. Bu döngünün döngü zamanı $10\epsilon + 10\delta + p_1 + p_2 + p_3 + p_4$ şeklindedir. Döngü zamanı incelendiğinde robotun herhangi bir makinede kısmi bekleme zamanı geçirmediği görülmektedir. Fakat birçok döngü için modelin çözümü bu kadar basit değildir. Makine sayısının artmasıyla beraber döngülerde robotun önünde kısmi bekleme yaptığı makine sayısı artmakta, bu da modeli karmaşıktırmaktadır. Döngülerden bir kısmı içinse, atama problemi NP-Zor' dur. Dolayısıyla, problemin çözümünde özellikle de makine sayısı ve işlem sayısı arttığında, sezgisel bir yöntemin kullanılması gerekmektedir. Bir sonraki bölümde çözüm için geliştirilen sezgiselden bahsedilecektir.

4. Sezgisel Çözüm Yöntemi

Bu bölümde, problemin çözümü için geliştirilmiş olan sezgisel çözüm yöntemi anlatılacaktır. Fakat bunun öncesinde, sezgisel algoritma açısından önemli bir yer tutan ve m makineli ve makinelerdeki işlem zamanları bilinen bir robotik hücrede en iyi 1-birim döngüsünü bulan Crama ve van de Klundert [19] algoritması anlatılacaktır.

4.1 Crama-van de Klundert [19] Algoritması

Daha önce de belirtildiği üzere m makineli bir sistemde $m!$ kadar 1-birim döngüsü vardır [72]. Fakat Crama ve van de Klundert [19], en iyi 1-birim döngüsünü bulmak için daha önce tanımlanan piramitsel permutasyonların incelenmesinin yeterli olacağını göstermişlerdir. Bunun sonucu olarak incelenmesi gereken alternatif döngü sayısı $m!$ 'den 2^{m-1} 'e inmiştir. Fakat bu sayı hala makine sayısına göre üstel olarak artış gösteren büyük bir sayıdır ve tek tek bütün alternatiflerin denenip en iyi 1-birim döngüsünün seçilmesi şeklindeki bir yaklaşımla makine sayısının büyük olduğu problemleri çözebilmek mümkün değildir. Bu sebeple, Crama ve van de Klundert [19] problemin polinom zamanlı çözümü üzerinde çalışmışlar ve aşağıda detayları anlatılan dinamik programlama algoritmasını geliştirmişlerdir.

Bu algoritmanın önemli bir alt rutininde, ϵ , δ , işlem zamanları ve herhangi bir piramitsel permutasyonun aktivite sıralaması verildiğinde, bu döngünün çevrim zamanı hesabı yapılabilmektedir. Bu açıdan, optimalin piramitsel permutasyonlar arasında olduğunun ispat edilmesi sadece alternatif döngü sayısını azaltmamış

aynı zamanda polinom zamanlı dinamik programlama algoritmasının geliştirilmesini de mümkün kılmıştır. Algoritmanın ana döngüsünde bu alt rutin sık sık kullanılmaktadır.

Algoritmanın ana döngüsü, en iyi piramitsel permutasyonu bulmayı hedeflemektedir. Bu kısımda genellik bozulmadan döngülerin A_0 aktivitesiyle başladığı varsayılmaktadır. Her 1-birim döngüsünde bu aktivite bir kez kullanıldığı için, bu varsayım genelliği bozmamaktadır. Dinamik programlama A_0 aktivitesiyle başlayan döngünün son aktivitesi olarak $A_1 \dots A_{m+1}$ aktivitelerini tek tek dener ve bu son aktiviteden bir sonraki çevrimin başında tekrar kullanılacak olan A_0 aktivitesine olan mesafeyi zaman cinsinden hesaplar. Daha sonra, bir önceki aktivitenin belirlenmesine geçilir. Tek tek bütün aktiviteler denenerek yine, bir sonraki çevrimin başına kadar geçecek minimum zamanlar hesaplanır. Bu işlem ilk aktivite olan A_0 'a ulaşana kadar devam eder. A_0 aktivitesine ulaşıldığında minimum çevrim zamanını veren aktivite sıralaması seçilir.

Geliştirilen bu dinamik programlama yaklaşımı $O(m^3)$ zamanda çalışan bir algoritmadır ve kesin çözümü vermektedir. Hatırlanacağı üzere, bu tezde ele alınan problemde hem robot hareket döngüsü hem de işlemlerin makinelere atanmasının belirlenmesi gerekmektedir. Crama ve van de Klundert [19] algoritması, işlem zamanları belli olduğunda, en iyi 1-birim döngüsünü vermektedir. Dolayısıyla, işlemlerin makinelere atanması bir şekilde bulunduğunda, buna karşılık gelen en iyi 1-birim döngüsünü bulmak için bu algoritma kullanılabilir. Fakat işlemlerin makinelere atanması problemi hala önemli bir problemdir ve bulunacak en iyi 1-birim döngüsü bu atamalara bağlıdır. Bir sonraki bölümde, işlemlerin makinelere atanmasını bulan ve Crama ve van de Klundert [19] algoritmasını bir alt rutin olarak kullanan sezgisel algoritmanın detaylarından bahsedilecektir.

4.2 Sezgisel Yöntem

Bu bölümde problemin çözümü için geliştirilmiş olan sezgisel algoritmanın detayları anlatılacaktır. Temel ilke olarak sezgisel algoritma ilk aşamada işlemlerin makinelere atanmasını belirlemekte, daha sonra bu atamalar sonucu ortaya çıkan işlem zamanlarını ve bir önceki bölümde anlatılan Crama ve van de

Klundert [19] algoritmasını kullanarak bu atamaya karşılık gelen en iyi 1-birim döngüsünü bulmaktadır. Fakat, işlemlerin makinelere atanması ve en iyi 1-birim döngüsünün bulunması birbirine bağımlı oldukları için, bu şekilde sıralı çalışan bir algoritmanın iyi çözümler bulma şansı düşüktür. Bu sebeple atamalar ve karşılık gelen döngü bulunduktan sonra en başa dönüp yeni bir atama ve ona karşılık gelen döngü bulunmakta, bu süreç belirli sayıda tekrar edilmekte ve içlerinden en iyi çözüm seçilmektedir.

Algoritmanın başarıya ulaşması için en önemli adımlardan bir tanesi işlemlerin makinelere atanmasıdır. Klasik montaj hattı dengeleme problemlerinde atamalar makinelerdeki işlem zamanları mümkün olduğunca birbirlerine eşit olacak şekilde yapılır. Böyle bir atama, ele alınan piramitsel permutasyonların bir kısmı için de en iyi çözümü verecektir. Örneğin, 4 makineli bir sistemdeki $A_0A_4A_3A_2A_1$ döngüsünün çevrim zamanı aşağıda (4.1) numaralı eşitlikte verildiği gibidir.

$$T_1^4 = 10\epsilon + 16\delta + w_1 + w_2 + w_3 + w_4 \quad (4.1)$$

$$\begin{aligned} w_1 &= p_1 - 6\epsilon - 12\delta - w_2 - w_3 - w_4 \\ w_2 &= p_2 - 6\epsilon - 12\delta - w_3 - w_4 \\ w_3 &= p_3 - 6\epsilon - 12\delta - w_1 - w_4 \\ w_4 &= p_4 - 6\epsilon - 12\delta - w_1 - w_2 \end{aligned}$$

Bu eşitlikten görüleceği üzere, bu döngü için en düşük çevrim zamanının elde edilebilmesi için makinelerdeki işlem zamanları birbirine eşit olmalıdır. Fakat, her piramitsel döngü için en iyi çözüm bu şekilde bir atamayla sağlanamayabilir. Örneğin, $A_0A_2A_4A_3A_1$ döngüsünün çevrim zamanı aşağıdaki eşitlik (4.2)' de verilmiştir.

$$T_4^4 = 10\epsilon + 14\delta + w_1 + w_2 + w_3 + w_4 \quad (4.2)$$

$$\begin{aligned} w_1 &= p_1 - 6\epsilon - 10\delta - w_2 - w_3 - w_4 \\ w_2 &= p_2 - 2\epsilon - 4\delta \\ w_3 &= p_3 - 2\epsilon - 4\delta - w_4 \\ w_4 &= p_4 - 6\epsilon - 10\delta - w_1 - w_2 \end{aligned}$$

Bu döngünün çevrim zamanı incelendiğinde, en iyi atamada, 3 ve 4. makinelere atanan işlem zamanları (p_3 ve p_4) mümkün olduğunca düşük olmalıdır. 2.

makinenin işlem zamanı (p_2) bunlardan daha fazla, 1. makinenin işlem zamanı (p_1) ise bunlardan daha fazla olabilir. Yani dengeli bir atama bu döngü için en iyi çözümü veremeyecektir.

Diğer taraftan bazı döngüler için de atama probleminin herhangi bir önemi yoktur. Örneğin, $A_0A_1A_2A_3A_4$ döngüsünün çevrim zamanı aşağıdaki eşitlik (4.3)'de verilmiştir.

$$T_8^4 = 10\epsilon + 10\delta + p_1 + p_2 + p_3 + p_4 \quad (4.3)$$

Bu döngü zamanı incelendiğinde, makinelerdeki işlem zamanları (işlemlerin makinelere atanması) ne olursa olsun çevrim zamanı değişmemektedir. Yani çevrim zamanı atamadan bağımsızdır. Bu döngülerin hepsi piramitsel döngü olduğu için, daha önce bahsedilen kurala göre, hepsinin en iyi olma ihtimali bulunmaktadır. ϵ , δ , işlem zamanları, öncelik ilişkileri ve atanabilirlik ilişkilerine bağlı olarak herhangi birisi en iyi olabilir. Bu özellikten dolayı, klasik montaj hattı dengeleme literatüründe kullanılan ve makinelerdeki işlem zamanlarını dengelemeye çalışan yöntemlerin bu çalışmada en iyi çözümü verme garantileri bulunmamaktadır. Bu sebeple, problemin çözümü için parametrik bir sezgisel algoritma geliştirilmiştir. Bu sezgisel yöntem, işlemleri makinelere atarken, makinelerdeki işlem zamanlarının belirli miktarda sapmasına izin vermektedir. Ne kadar sapma yapacağı ise bazı parametrelerle belirlenmektedir.

Makinelerdeki işlem zamanlarının nasıl değiştiğini daha detaylı anlatmak gerekirse, makinelerdeki işlem zamanlarının büyüklüklerinin sinüzoidal bir eğri çizdiği varsayılmıştır. Geliştirilen sezgiselin de 3 farklı parametresi bulunmaktadır. Bu üç parametreden ilki α , makinelerin işlem zamanlarının ortalama işlem zamanından sapma miktarının büyüklüğünü göstermektedir. α 'nın 0 olması makinelerin işlem zamanlarında herhangi bir sapma olmaması, yani bütün makinelerin işlem zamanlarının birbirine eşit yapılmaya çalışılması anlamına gelmekteyken, α 'nın değerinin büyük olması sapma miktarının da büyük olması anlamına gelmektedir. Kullanılan parametrelerden ikincisi β 'dir. Bu parametre ise işlem zamanlarının sinüs eğrisinin hangi noktasından başladığını ifade etmektedir. Örneğin, sinüs eğrisinin 0 olduğu noktadan, 1 olduğu noktadan, -1 olduğu noktadan veya bunlar arasında kalan herhangi bir noktadan başlayabilir. β parametresi derece cinsinden bir ifadedir. Sezgiselin son parametresi ise γ parametresidir. Bu

parametre de ardışık iki makinenin işlem zamanlarının sinüs eğrisinde hangi noktalara denk geldiğini, yani aralarındaki açıklığı ifade etmektedir. Anlatılan bu parametreler için değişik değerler kullanarak Şekil 4.1'de çeşitli örnekleri verilen işlem zamanları grafikleri elde edilebilir.

Klasik montaj hattı dengeleme problemlerinde kullanılan bazı yöntemlerde (Örneğin, pozisyon ağırlığı yöntemi, aşamalı sıralama yöntemi v.s.) istenen çevrim zamanları belirlenmekte, daha sonra da işlemlerin makinelere atanması belirlenirken, bu çevrim zamanı aşılmayacak şekilde makinelere atanması yapılmaktadır. Fakat bu yöntemlerde, her makinede kullanılan limit aynıdır. Bu çalışmada ele alınan problemde ise yukarıda bahsedildiği gibi makinelerin işlem zamanları arasında belirli miktar sapmaya izin verilmesi gerekmektedir. Bunu sağlamak için her makineye aynı limit uygulamak yerine bu çalışmada her makine için ayrı bir limit kullanılmıştır. Bu limitlerin hesaplanmasında ise yukarıda anlatılan α , β ve γ parametreleri kullanılmıştır.

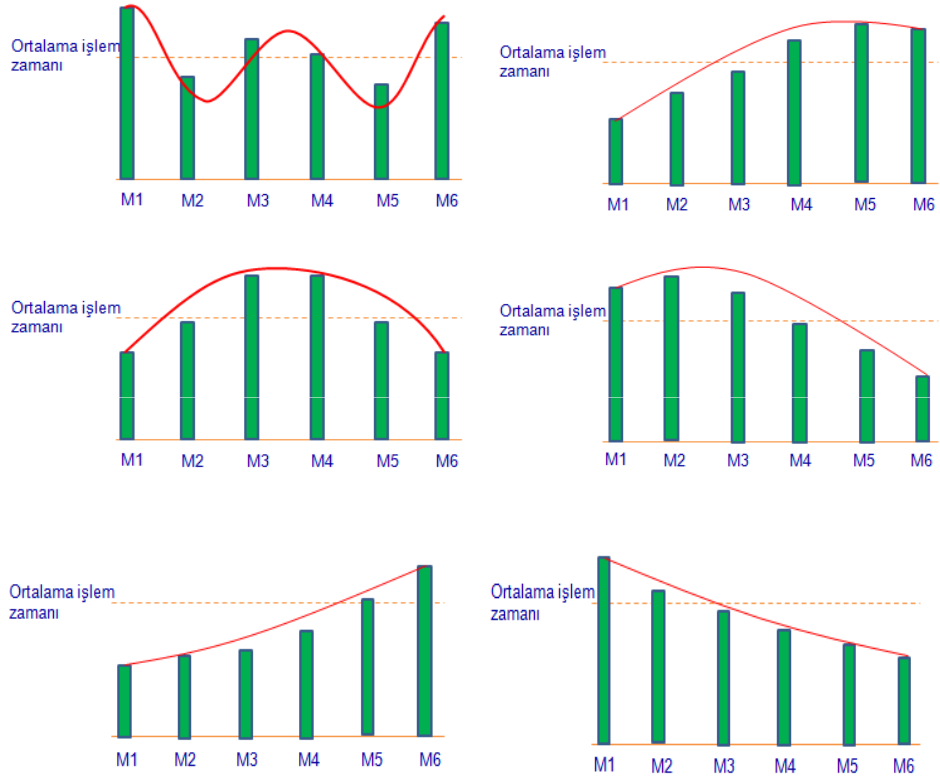
Herhangi bir makine için işlem zamanları limiti U_i , (4.4)'de verilen formüle göre hesaplanmaktadır. Bu sezgiselde tek bir α , γ , β değeri kullanılmamakta, değişik değerler için problem tekrar tekrar çözdürülmekte ve en iyi çözüm seçilmektedir. Sezgisel boyunca kullanılan α değerleri (4.5)'teki formüle göre hesaplanmaktadır. Burada kullanılan α_k , k'inci tekrarda α için kullanılan değeri göstermektedir. γ değerleri ise (4.6)'teki formüle uygun değerler almaktadır. γ_l , l'ninci tekrarda γ için kullanılan değerdir. β ise belirlenen bir üst limite göre artış göstermektedir. Örneğin, üst limit 180, 270 ya da 360 derece gibi değerler alabilir. Artış miktarı ise yine farklı derecelerde olabilir. Makinelerin üst limitleri hesaplanırken her makineye düşen ortalama işlem zamanı kullanılacaktır. R , her bir makineye düşen ortalama işlem zamanını gösterecektir.

$$U_i = R + \alpha \sin(\beta \cdot i\gamma)\pi \quad (4.4)$$

$$\alpha_k = R - \frac{(k \cdot R)}{10} \quad (4.5)$$

$$\gamma_l = \frac{45 \cdot l}{m - 1} \quad (4.6)$$

Fakat, herhangi bir parametre için hangi parametre değerinin en iyi sonucu vereceği bilinmemektedir. Bu sebeple, her problem için farklı α , β ve γ değerleri denenerek içlerinden en iyi sonucu veren seçilmektedir. Kaç değişik α değerinin



Şekil 4.1: Sinüs grafiğine göre işlem zamanları

kullanılacağı, β ve γ ' nin hangi aralıkta ve nasıl değerler alacağıyla ilgili olarak bir sonraki bölümde detaylı bir kalibrasyon çalışması yapılmıştır.

Makine limitlerinin nasıl belirlendiği anlatıldıktan sonra geliştirilen algoritmanın adımları aşağıdaki gibi özetlenebilir:

Girdi: $m, N, a_{ij}, b_{jk}, o_j$

Çıktı: İşlemlerin makinelere atanması, en iyi 1-birim döngü ve çevrim zamanı

1. Değişik α, β ve γ değerleri için tekrar et.

1.1 Her makine için üst sınırı bul. M_i için üst sınır:

$$U_i = R + \alpha \sin(\beta \cdot i\gamma)\pi \text{ (Eşitlik 4.4)}$$

1.2 Önceliği olmayan işlerden işlem zamanı en büyük olanı atanabileceği ilk makineye ata. İşlemin atanabileceği makineyi belirlerken atanabilirlik matrisini ve makineye o ana kadar atanmış işlerle beraber yeni işin işlem zamanları toplamının makine işlem zamanı üst limitini geçmemesini kontrol et. Atama

yapıldıktan sonra, atanan işin ardılı olan işlerin atanabilirlik matrisini bu işin atandığı makinelerden daha öncekileri silerek güncelle, atanan işi öncelik listesinden sil. Makinedeki toplam işlem zamanını güncelle.

- 1.3 Bütün işlemler atandıysa, makinelerdeki işlem zamanlarını hesapla. Crama ve van de Klundert [19] algoritmasıyla çevrim zamanını hesapla ve diğer parametreler için hesaplanmış olan çevrim zamanlarıyla kıyasla. Değilse Adım 1.2' ye geri dön.

Algoritma 1 Robotik Hat Dengeleme Algoritması (ROHDA)

Girdi: $m, N, a_{ij}, b_{ij}, o_i, \epsilon, \delta \quad \forall i, j$

Çıktı: C_{max}

minimum çözüm=M

for $\alpha = 1; \alpha \leq 10; \alpha ++$ **do**

for $\beta = 0; \beta \leq 180; \beta + 30$ **do**

for $\gamma = 45; \gamma \leq 360; \gamma + 45$ **do**

$mtop_i = 0 \quad \forall i$

$U_i = R + \alpha \cdot \sin(\beta + i \cdot \gamma)$ (Eşitlik 4.4)

for $l = 0; l \leq N; l ++$ (bütün işler için) **do**

for $k = 0; k \leq N; k ++$ (öncelik matrisi için) **do**

$y_k = 0$ önceliği olmayan ilk işi bul.

for $i = 0; i \leq m; i ++$ (atama matrisi için) **do**

$a_{ij} = 1$ ve $mtop_i + o_j \leq U_i$ ise atama yap.

$mtop_i = mtop_i + o_j$ bütün makineler için işlem süresi hesapla.

for $j = 0; j \leq N; j ++$ **do**

$b_{kj} = 0$ atanan işi öncelik matrisinden kaldır ve atanabilirlik matrisini ardıl işlere göre güncelle.

end for{j}

end for{i}

end for{k}

end for{l}

 Crama algoritmasını çalıştır.

if yeni çözüm \leq minimum çözüm **then**

 minimum çözüm=yeni çözüm

end if

end for{ γ }

end for{ β }

end for{ α }

C_{max} ve döngü çıktısı olarak al.

Geliştirilen algoritma makine sayısı, işlem sayısı, işlem zamanları, atanabilirlik ve öncelik matrisleri, ϵ ve δ değerlerini girdi olarak kullanır. Daha sonra makine işlem

zamanları için üst sınır belirlemek üzere α , β ve γ parametrelerini ait belirlenen aralıkların alt sınırından itibaren döngüleri başlatır. Üst sınırlar bütün makineler için ayrı ayrı belirlenir. Bütün işler için, öncelik matrisini kontrol ederek, önceliği olmayan ilk işi bulur. Bu işin atamasını gerçekleştirir ve atadığı makinenin işlem zamanına bu işin işlem zamanını ekler. Atanan iş öncelik matrisinden kaldırılır ve atanabilirlik matrisi güncellenir. Algoritma tüm işleri makinelere atadıktan sonra, Crama ve van de Klundert [19] algoritmasını çalıştırır. Çıktı olarak ise, en iyi 1-birim döngüsünü ve çevrim zamanını verir.

ROHDA'yı bir örnekle açıklayalım.

Örnek 1: Bu örnek daha sonra performans testlerinde de kullanacağımız çevrimiçi kütüphaneden [77] alınan Buxey örneğidir. Problem 29 işlemden oluşmaktadır ve işlemlere ait işlem zamanları Tablo 4.1'deki gibidir. Bu probleme ait öncelik ilişkileri Tablo 4.2'de ve atanabilirlik matrisi ise Tablo 4.3'tedir. İşlem sayısı, işlem zamanları, atanabilirlik matrisi ve öncelik ilişkileri sezgisel algoritmanın girdilerini oluşturmaktadır. Bunların dışında makine sayısı, ϵ ve δ değerleri de belirlenmelidir. Bu örnek için makine sayısı 5, ϵ ve δ değerleri ise sırasıyla 2 ve 3 alınmıştır.

İşlem No	İşlem Süresi	İşlem No	İşlem Süresi	İşlem No	İşlem Süresi
1	7	11	21	21	1
2	19	12	10	22	9
3	15	13	9	23	25
4	5	14	4	24	14
5	12	15	14	25	14
6	10	16	7	26	2
7	8	17	14	27	10
8	16	18	17	28	7
9	2	19	10	29	20
10	6	20	16		

Tablo 4.1: Örnek 1'deki problem için işlem zamanları

Girdi olarak belirlenen tüm parametrelerden sonra, algoritma üst sınır için belirlenen α , β ve γ parametrelerinin alt limitlerinden başlayarak, üst limitlerine kadar devam edecek olan bir döngüye girer. Böylelikle bu döngüde makine üst limitleri sürekli değişeceğinden optimal çözüme yaklaşma ihtimali de artacaktır.

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

Tablo 4.2: Örnek 1' deki problem işlemlerinin öncelik ilişkileri b_{kj}

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

	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	1	0	1	1	0	0	0	0	0	1	1	1	0	0
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	0	0	0	0	1	1	1	1	0	0
5	0	0	0	0	0	0	0	0	1	1	1	1	0	1

Tablo 4.3: Örnek probleme ait atanabilirlik matrisi a_{ij}

Daha sonra bütün işler arasından, öncelik matrisinde önceliği olmayan işler arasından işlem zamanı en büyük olanı seçer. İlk seçilen iş 7. iştir. Atanabilirlik matrisini kontrol eder, bu işin atanabileceği ilk makineyi bulur ve eğer işin ataması gerçekleşirse makine işlem zamanı üst limitinin aşılmadığını kontrol eder. Aşılmazsa atamayı gerçekleştirir. Buradaki ilk seçilen iş 7. iş olduğundan ve atanabilirlik matrisinde 7. işin atanabileceği ilk makine 1. makine olduğundan, daha önce atama yapılmadığı için önceliği olmayan en büyük işlem zamanına sahip iş bu makineye atanabilir. Ataması gerçekleştirilen 7. iş öncelik matrisinden kaldırılır ve atama matrisi güncellenir. Daha sonra 25. iş önceliği olmayan işler arasından işlem zamanı en büyük olan olduğu için seçilir, atama matrisi ve atanabilirdiği ilk makine üst sınırı kontrol edilerek ataması gerçekleştirilir. 25. iş, 1. makineye atanmıştır. Sonra tekrar önceliği olmayan işler arasından en büyük işlem zamanlı iş bulunur. Bu iş, 7. işin ataması gerçekleştiği ve öncelik matrisinden silindiği için, 12. iştir. 12. işin de atamasının gerçekleşmesi için atanabilirlik matrisi ve makine üst sınırı kontrol edildikten sonra ataması gerçekleştirilir. 12. iş 1. makineye atanmıştır. Bu şekilde tüm işlerin ataması gerçekleştirildikten sonra bulunan makine işlem zamanları Tablo 4.4'teki gibidir. Daha sonra her bir α , β , γ parametresi için Crama ve van de Klundert [19] çağırılır ve çevrim zamanı bulunur. Bulunan çevrim zamanı eski çevrim zamanı ile karşılaştırılır, küçük değere sahip olan seçilir ve en iyi 1-birim döngü ile çevrim zamanı bulunmuş olur. Bu problem için bulunan en iyi 1-birim döngü $A_0A_4A_5A_3A_2A_1$ şeklindedir ve bu döngüye ait çevrim zamanı 116' dir.

Makine	Atanan İşler										Toplam İşlem Zamanı
1	1	2	3	4	6	7	9	10	12	25	96
2	5	8	11	13	17	20	26				90
3	14	15	16	18	19	21	22	23	28		94
4	24	27									24
5	29										20

Tablo 4.4: Örnek 1' de ROHDA'ya göre oluşan makine işlem zamanları

ROHDA'nın optimal çözüm verip vermediğini kontrol etmek için, aynı verilerle matematiksel model çözdürülür. m makineli bir sistemde daha önce de belirtildiği gibi 2^{m-1} tane piramitsel permutasyon bulunmaktadır ve optimal çözüm bu piramitsel permutasyonlardan biridir. Çözdürülen matematiksel modelde 5

Makine	Atanan İşler										Toplam İşlem Zamanı
1	1	2	3	4	5	7	9	10	12	14	88
2	6	8	11	13	15	16	19	21			98
3	17	18	20	22	23	28					88
4	24	25	26	27							40
5	29										20

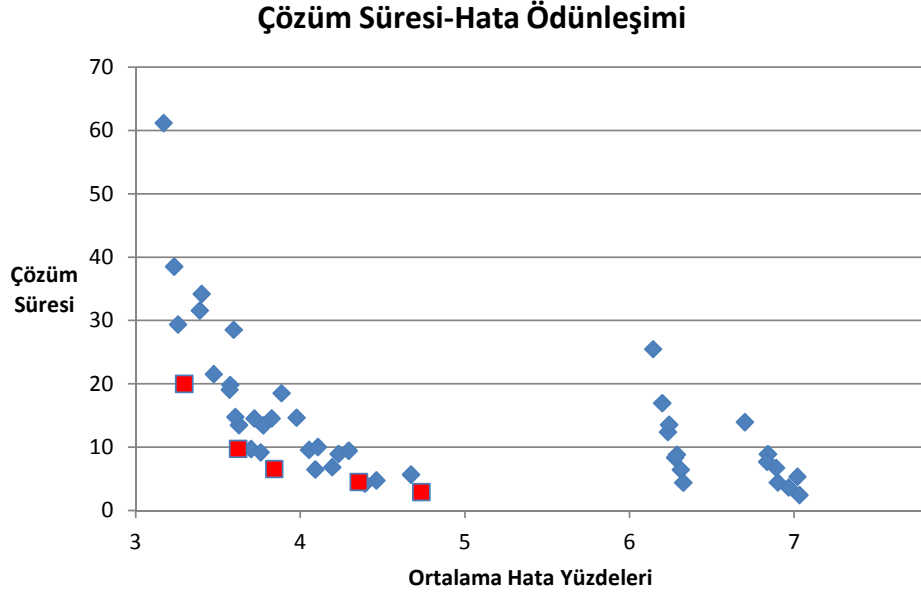
Tablo 4.5: Örnek 1' de GAMS' e göre oluşan makine işlem zamanları

makine için 16 tane piramitsel permutasyon bulunduğundan bu permutasyonların parametreler göre çevrim zamanı hesaplanır. En düşük çevrim zamanını veren döngü optimal döngüdür. Bu örnek için GAMS modeli çözdürüldüğünde, $A_0A_4A_5A_3A_2A_1$ döngüsünü en iyi 1-birim döngü olarak bulmuştur. Optimal çevrim zamanı ise 108 çıkmıştır ve işlerin makinelere atanması Tablo 4.5'teki gibidir.

ROHDA ile GAMS modeli karşılaştırıldığında, %6,89 luk bir hata payı olduğu görülmektedir. İşlerin makinelere atanmasındaki farklılık bu hata payını ortaya çıkarmaktadır.

4.2.1 α , β ve γ Parametrelerinin Belirlenmesi

α , β ve γ parametreleri sezgisel algoritma geliştirilirken makine üst sınırlarının belirlenmesinde kullanılmıştır. Üst sınırlar, bu parametrelere göre değişiklik gösterir. ROHDA, değişik α , β , γ değerleri kullanarak en iyi sonuca ulaşmaktadır. Bu parametreler için her çalıştırmada çok sayıda değer kullanılması çözüm kalitesini artırırken çözüm zamanını uzatmaktadır. Az sayıda değer kullanılması ise tam tersi bir etkiye sahiptir. Bu parametreler için kullanılacak değer sayısını ve hangi değerlerin kullanılacağını belirlemek amacıyla bu çalışmada bir kalibrasyon çalışması yapılmıştır. Buna göre 6 makineli bir sistem ele alınmış, çevrimiçi kütüphaneden [77] alınan 17 problem, 3'er farklı ϵ ve δ değeri kullanılarak ve her problem için 5 replikasyon yapılarak hem ROHDA hem de GAMS ile çözdürülmüştür. α parametresi, 1 – 10, 1 – 5 ve 5 – 10 arasında 3 farklı değer alabilirken, β ve γ değerleri farklı dereceler alabilir. β parametresi 30 yada 45 derecelik artışlarla 180 yada 360 dereceye kadar ilerleyebilir. γ parametresi ise 30 yada 45 derecelik artışlarla 180 yada 360 dereceye kadar ilerleyebilir. Bu



Şekil 4.2: Çözüm süresi-hata ödünleşimi

aralıklardan yola çıkarak 48 farklı durum incelenmiştir.

$$\text{Ortalama hata} = 100 \left(\frac{\text{ROHDA Sonuç} - \text{GAMS Sonuç}}{\text{GAMS Sonuç}} \right)$$

sonucuna göre hesaplanmaktadır. Her bir durum üzerine ortaya çıkan ortalama zaman ve ortalama hatalar belirlenmiştir. Bu parametrelerin değişimine göre ortaya çıkan sonuçlar Tablo 4.6'deki gibidir.

Tablo 4.6'den yola çıkarak α , β ve γ parametrelerin alabileceği değerlere göre tüm durumlardan elde edilen verilerle genel ortalama hata ve ortalama çözüm süreleri değerleri hesaplanmıştır. Parametrelere göre değişen bu değerler Tablo ??'deki gibidir. Sezgiselde kullanılacak α , β ve γ 'nın seçiminde çözüm kalitesi kadar zamanı da önemli bir kriterdir. Örneğin, Tablo 4.6 incelendiğinde en düşük hata ortalaması α 1-10 aralığında, β 360 derece limite kadar 30 derecelik artışlarla ve γ da yine 360 dereceye kadar 30 derecelik artışlarla kullanıldığında elde edilmektedir. Bu değerler için ortalama hata %3,17'dir. Fakat bu çözüme ulaşmak için gerekli ortalama zaman 61.19 saniyedir. Makine sayısı arttıkça da bu çözüm süresi artış gösterecektir. Diğer taraftan α 5-10 aralığında, β 180 derece limitle 60 derecelik artışlarla ve γ 180 derece limitle 45 derecelik artışlar kullanıldığında çözüm zamanı 2.45 saniye gibi çok düşük bir değer elde

α Aralık	β		γ		Ortalama Hata %	Ortalama Süre (sn)
	Üst Limit	Artış	Alt Limit	Artış		
(1-5)	180	30	30	6	4,09	6,48
(1-5)	180	30	30	12	3,72	14,54
(1-5)	180	30	45	4	4,46	4,75
(1-5)	180	30	45	8	4,05	9,57
(1-5)	180	60	30	6	4,39	4,27
(1-5)	180	60	30	12	4,11	10,03
(1-5)	180	60	45	4	4,74	2,90
(1-5)	180	60	45	8	4,67	5,68
(1-5)	360	30	30	6	3,83	14,54
(1-5)	360	30	30	12	3,59	28,54
(1-5)	360	30	45	4	4,23	8,94
(1-5)	360	30	45	8	3,89	18,53
(1-5)	360	60	30	6	4,19	6,84
(1-5)	360	60	30	12	3,98	14,66
(1-5)	360	60	45	4	4,36	4,50
(1-5)	360	60	45	8	4,29	9,45
(5-10)	180	30	30	6	6,89	6,74
(5-10)	180	30	30	12	6,7	13,97
(5-10)	180	30	45	4	6,9	4,43
(5-10)	180	30	45	8	6,84	8,91
(5-10)	180	60	30	6	6,97	3,65
(5-10)	180	60	30	12	6,84	7,7
(5-10)	180	60	45	4	7,03	2,45
(5-10)	180	60	45	8	7,02	5,34
(5-10)	360	30	30	6	6,23	12,40
(5-10)	360	30	30	12	6,14	25,49
(5-10)	360	30	45	4	6,28	8,36
(5-10)	360	30	45	8	6,20	16,95
(5-10)	360	60	30	6	6,31	6,45
(5-10)	360	60	30	12	6,24	13,54
(5-10)	360	60	45	4	6,33	4,39
(5-10)	360	60	45	8	6,29	8,84
(1-10)	180	30	30	6	3,63	13,50
(1-10)	180	30	30	12	3,26	29,38
(1-10)	180	30	45	4	3,7	9,71
(1-10)	180	30	45	8	3,3	19,99
(1-10)	180	60	30	6	3,76	9,19
(1-10)	180	60	30	12	3,47	21,54
(1-10)	180	60	45	4	3,84	6,55
(1-10)	180	60	45	8	3,77	13,46
(1-10)	360	30	30	6	3,4	34,20
(1-10)	360	30	30	12	3,17	61,19
(1-10)	360	30	45	4	3,57	19,85
(1-10)	360	30	45	8	3,23	38,53
(1-10)	360	60	30	6	3,61	14,78
(1-10)	360	60	30	12	3,39	31,59
(1-10)	360	60	45	4	3,62	9,72
(1-10)	360	60	45	8	3,57	19,07

Tablo 4.6: α , β ve γ Parametrelerinin değişik değerleri ile elde edilen sonuçlar

etmekteyken ortalama hata %7.03 olmaktadır. Dolayısıyla her iki kriter arasından da daha iyi sonuçlar veren parametre değerinin seçilmesi gerekmektedir. Bunun için Şekil 4.2'de görülen çözüm süresi- hata ödünleşim grafiği faydalı olacaktır. Bu grafiğe göre birçok nokta diğerlerinden hem çözüm süresi hem de çözüm kalitesi açısından daha kötüdür, yani başatlanmıştır. Başatlanmayan çözümler arasından ise alternatif olabilecek beş tanesi grafikte işaretlenmiştir. Bu beş çözüm incelendiğinde, ortalama hata değeri %3.3, çözüm zamanı da 19.99 sn olan parametre değerleri en ideal değerler olarak bu çalışmada seçilmiştir. Buna göre, ROHDA' da kullanılacak parametre değerleri α 1 – 10 aralığında, β 30 derecelik artışlarla 180 dereceye kadar ve γ 45 derecelik artışlarla 360 dereceye kadar olduğu durum seçilmiştir. Buna göre ROHDA'da kullanılan bütün α , β , γ değerleri Tablo 4.7'deki gibidir.

α	β	γ
1	30	45
2	60	90
3	90	135
4	120	180
5	150	225
6	180	270
7		315
8		360
9		
10		

Tablo 4.7: α , β ve γ değerlerinin değişimi

5. Deneysel Çalışma

Bu bölümde sezgisel yöntemin çözüm kalitesinin testi için yapılan deneysel çalışma, elde edilen sonuçlar ve değerlendirmeler sunulmuştur.

5.1 Deneysel Çalışma Parametreleri

Sezgisel çözüm yönteminin kalitesi problem girdisi olan parametreler için değişik değerler kullanılarak test edilmiştir. Makine sayısı (m), ϵ , δ , işlem sayısı (N), işlem zamanı (o_j), öncelik ilişkileri (b_{jk}) ve atanabilirlik ilişkileri (a_{ij}) kullanılan parametrelerdir. Bu parametrelerden bazıları türetilmiş, bazıları ise daha önce literatürde kullanılmış çevrimiçi kütüphaneden [77] parametrelerdir. Makine sayısı, işlem sayısı, işlem zamanı ve öncelik ilişkileri literatürde daha önce kullanılan test verilerindedir. ϵ ve δ değerleri ile atanabilirlik matrisi çevrimiçi kütüphanede [77] yer almadığından bu çalışma kapsamında türetilmiştir.

Döngü sayıları daha önce de belirtildiği gibi makine sayısına bağlı olarak artış göstermektedir. Crama ve van de Klundert [19] piramitsel permutasyonların optimal olduğu şeklindeki sonuçları kullanılsa bile ele alınacak döngü sayısı makine sayısına göre hızla artmaktadır. Bu sebeple daha yüksek makineli problemler için optimal çözümü Bölüm 3'te anlatılan matematiksel model kullanılarak uygun bir süre içerisinde bulmak mümkün olmadığından makine sayısı 4, 5 ve 6 olarak 3 farklı seviye ile sınırlandırılmıştır. Dolayısıyla 4 makineli bir sistem için optimal çözümün bulunmasında çözdürülecek model sayısı 8, 5 makineli bir sistem için 16 ve 6 makineli bir sistem için 32' dir.

Geliştirilen sezgiselin performansını etkileyeceği düşünülen diğer önemli parametreler ise parçaların sahip olduğu işlem sayılarıdır. İşlem sayısının düşük olması atama problemini kolaylaştırırken fazla olması problem boyutunu büyütecektir. Diğer bir önemli faktör ise bu işlemlerin işlem süreleridir. İşlem sürelerinin küçük veya büyük olması veya işlem süreleri arasındaki varyansın fazla olmasını problemin karmaşıklığını ve geliştirilen sezgiselin performansını etkileyeceği düşünülmektedir. İşlemler arası öncelik ilişkilerinin de benzer etkileri olacaktır. Öncelik ilişkilerinin sıkı olması veya daha gevşek olması çözüm zamanını ve kalitesini etkileyebilecektir. Bu 3 parametre montaj hattı dengeleme için kullanılan çevrimiçi kütüphaneden [77] elde edilmiştir. Basit hat dengeleme probleminin çevrim zamanının enküçüklenmesi (SALBP-2) amaç fonksiyonuna dair belirlenen veriler ele alınmıştır. Bu veri setinde 17 farklı problem ve bu problemlere bağlı işlem sayısı, işlem zamanları ve öncelik ilişkileri mevcuttur. Her problem için işlem sayıları, işlem zamanları ve öncelik ilişkileri farklılık göstermektedir. Problemlerin literatürdeki isimleri ve işlem sayıları Tablo 5.1'de verilmiştir.

Problem No	Problem Adı	İşlem Sayısı
1	Buxey	29
2	Gunther	35
3	Kilbridge	45
4	Lutz1	32
5	Lutz2	89
6	Sawyer	30
7	Tonge	70
8	Arcus1	83
9	Arcus2	111
10	Hahn	53
11	Warnecke	58
12	Wee-mag	75
13	Lutz3	89
14	Mukherje	94
15	Barthold	148
16	Barthold2	148
17	Scholl	297

Tablo 5.1: Problem İsimleri ve İşlem Sayıları

Sezgiselin performansını etkileyebileceği düşünülen bir diğer önemli faktör ise atanabilirlik ilişkileridir. Her işlemin her makineye atanabildiği tam esnek sistemlerle, atanabilirliğin kısıtlandığı daha sıkı sistemlerde sezgiselin performansı

farklılık gösterecektir. Bu sebeple, atanabilirlik açısından bu iki farklı durum ele alınmıştır. Literatürde veya çevrimiçi kütüphanelerde bu çalışmada ele alındığı şekliyle bir atanabilirlik kısıtı veya verileri bulunmamaktadır. Bu sebeple, atanabilirlik ilişkileri bu çalışma kapsamında türetilmiştir. Bu türetme işlemi yapılırken atanabilirlik ilişkilerini öncelik ilişkilerinden bağımsız olarak düşünmek mümkün değildir. Aksi takdirde, ortaya çıkan problem mümkün olmayan bir probleme dönüşebilir. Bunu engellemek için şu kurallar geliştirilmiştir:

Kural: a işi b işinin önceliği ise (Bu öncelik hemen öncelik olmak zorunda değildir, a işi dolaylı yoldan b işinin önceliği olsa bile bu kural geçerli olmalıdır.); a işinin atanabileceği ilk makine b işinin atanabileceği son makineden daha önce olmalıdır.

Kural olmadığı takdirde mümkün bir çözüme ulaşmak mümkün değildir. Sağlandığında ise mümkün bir çözüm bulma garantisi vardır. Fakat aşağıdaki iki durum da gereksiz atanabilirlikleri ifade etmektedir ve bu durumda yazan atanabilirliklerin matristen silinmesinin herhangi bir zararı bulunmamaktadır.

- 1: a işinin atanabileceği ilk makine, b işinin atanabileceği ilk makineden aynı veya daha sonra olmamalıdır.
- 2: a işinin atanabileceği son makine, b işinin atanabileceği son makineden aynı veya daha sonra olmamalıdır.

Bu kuralların işleyişi bir örnekle açıklanacak olursa, 5 makineli bir robotik hücre ele alalım. Yukarıda belirtildiği gibi a işi b işinin önceliği olsun. a işinin ve b işinin iki farklı makineye atanabileceği varsayımı altında kurallardan yola çıkarak atama işlemi incelendiğinde, Şekil 5.1' deki matrisler elde edilebilir. a işinin 3. ve 4. makineye atanabileceği varsayılırsa, b işinin kurallar doğrultusunda 3 farklı atama durumu gerçekleşmez. Durum 1' de b işi 1 ve 2 numaralı makinelere atanmıştır. Fakat bu durumda a işinin ilk makinesi b işinin son makinesinden daha sonradır. Bu yüzden bu durumdaki atamalar geçerli değildir. Durum 2'de b işi 2, 3 ve 4 numaralı makinelere atanmıştır. Kurallar incelendiğinde b işinin hiç bir zaman 2 numaralı makineye atanamayacağı görülmektedir. Durum 3'te ise b işi 3 ve 4

	1	2	3	4	5	
a			1	1		Durum 1
b	1	1				

	1	2	3	4	5	
a			1	1		Durum 2
b		1	1		1	

	1	2	3	4	5	
a			1		1	Durum 3
b			1	1		

Şekil 5.1: Atanabilirlik kuralları doğrultusunda atanabilirlik matrisinin oluşturulması

numaralı makineler ve a işi 3 ve 5 numaralı makineler atanmıştır. Bu durumda da a işinin hiç bir zaman 5 numaralı makineye atanmayacağı görülmektedir.

Atanabilirlik matrisi oluşturulurken, matristeki 0'ların ve 1'lerin yoğunluğuna göre iki farklı seviye belirlenmiştir. Bu yoğunluğu belirtmek için Dar-el [21]'in geliştirdiği f-ratio değeri kullanılmıştır. F-ratio değeri, matristeki tüm 0 sayısının, matris eleman sayısına bölünmesiyle elde edilmektedir.

$$F - ratio = \frac{\text{Matristeki 0'ların sayısı}}{\text{Matristeki toplam eleman sayısı}}$$

F-ratio değeri göz önüne alınarak, atanabilirlik matrisinin oluşturulması için oluşturulan matris öncelik ilişkilerine bağlı olarak yukarıda bahsedilen kurallara uyması için bir algoritma geliştirilmiştir. Algoritma adımları aşağıdaki şekilde ifade edilebilir.

Girdi: $N, m, F - ratio, b_{kj}, \forall k, j$

Çıktı: $a_{ij} \quad \forall i, j$

Adım 1: F-ratio değerini gözeterek rastgele bir matris türet.

Adım 2: Bütün işlemler için, matrisin mümkünlük kuralına uyup uymadığını kontrol et. Uyarsa, DUR. Değilse Adım 1'e dön.

Bu çalışmada F-ratio için iki farklı seviye kullanılmıştır. Bunlardan ilki her işin her makineye atanabildiği tam esnek bir sistemdir (F-ratio=1). Diğeri ise atanabilirliklerin limitlendiği bir sistemdir. Bu limitli sistem için F-ratio değeri *Uniform*(0.4,0.5) olarak türetilmiştir.

Ek 1' de verilen çevrim zamanları incelendiğinde ϵ ve δ değerlerinin işlem zamanlarına göre büyüklüğü problemin karmaşıklığını etkileyebilmektedir. Bu sebeple her iki parametre için de 3 farklı değer ele alınmıştır. Bu değerler düzgün dağılımdan aşağıdaki aralıklar kullanılarak hesaplanmıştır:

- $uniform\left(0, \frac{R}{50}\right)$
 - $uniform\left(\frac{R}{50}, \frac{R}{30}\right)$
 - $uniform\left(\frac{R}{30}, \frac{R}{15}\right)$
-
- $uniform\left(mak\left\{0, (m-2)\frac{R-(m-2)\epsilon}{100}\right\}, \frac{R-(m-2)\epsilon}{50}\right)$
 - $uniform\left(mak\left\{0, (m-2)\frac{R-(m-2)\epsilon}{50}\right\}, \frac{R-(m-2)\epsilon}{25}\right)$
 - $uniform\left(mak\left\{0, (m-2)\frac{R-(m-2)\epsilon}{25}\right\}, \frac{R-(m-2)\epsilon}{15}\right)$

Parameters	Level 1	Level 2	Level 3
m	4	5	6
epsilon	düşük	orta	yüksek
delta	düşük	orta	yüksek
F-ratio	0	(0.4-0.5)	

Tablo 5.2: Bütün parametreler ve kullanılan değerler

Sonuç olarak, parametreler ve bunlar için kullanılan seviyeler Tablo 5.2'deki gibidir. Toplamda çevrimiçi kütüphaneden alınan 17 farklı problem için Tablo

5.2 verilen parametre seviyelerinin bütün kombinasyonları kullanılmış ve her kombinasyondan 5 tekrar alınmıştır. Toplamda $17 \times 3 \times 3 \times 3 \times 2 \times 5 = 4590$ farklı örnek problem ele alınmıştır.

5.2 Sonuçların Analizi

Performans testi için yapılan bütün çalışmaların detaylı sonuçları Ek A'da ve Ek B'de verilmiştir. Bütün sonuçlar Tablo 5.2'deki gibi özetlenebilir. 4, 5 ve 6 makinelerin her biri için 1530 farklı deneme yapılmıştır. Bu denemeler sonucunda Tablo 5.2' de görüldüğü gibi 4 makineden 5 ve 6 makineye geçişte ortalama yüzde hata miktarında artış gözlemlenmiştir. Algoritmanın çözüm süresinin ise makine sayısından etkilendiği söylenemez fakat GAMS çözüm süresinde 3 makineden 4 makineye geçişte yaklaşık olarak 3 kat bir artış söz konusudur. 6 makineye geçişte ise bu değer giderek artmakta ve 4 makineye göre yaklaşık 12 katına, 5 makineye göre ise yaklaşık 4 katı kadar bir değer almaktadır.

		Ortalama	GAMS	ROHDA
		Hata %	CPU (sn)	CPU (sn)
m	4	2,22	3,09	17,25
	5	5,97	8,67	17,93
	6	5,21	35,43	17,60
	Toplam	4,47	15,73	17,59

Tablo 5.3: Makine sayısına göre

Buradan da gözlemleneceği gibi, makine sayısı artışı ROHDA çözüm süresini etkilemezken, GAMS çözüm süresini ciddi bir oranda artırmaktadır. Bu durum ROHDA'nın özellikle makine sayısının büyük olduğu durumlarda etkin bir çözüm yöntemi göstermektedir.

Problemler ayrı ayrı olarak ele alındığında, her problem için makine sayısına göre değişimlerle beraber ortalama yüzde hata, GAMS ve ROHDA çözüm süreleri incelenmiştir. 17 farklı problem doğrultusunda elde edilen sonuçlar Tablo 5.2' teki gibidir. İşlem sayısının fazla olduğu problemlerde çözüm süreleri de artış göstermiştir. Örneğin, ilk problem 29 işlem sayısına sahiptir ve GAMS için ortalama çözüm süresi 2,71 sn iken ROHDA için çözüm süresi 3,65 sn'dir. 17. problem de ise işlem sayısı 297'dir. GAMS çözüm süresi 48,3 sn iken algoritma için

çözüm süresi 92, 61'dir. GAMS çözüm süresinin en fazla 8. problemde olduğu buna karşılık gelen ROHDA çözüm süresinin ise GAMS çözüm süresine göre çok daha düşük olduğu görülmüştür. En yüksek ROHDA çözüm süresine sahip problemin ise, 17. problem olduğu ve GAMS çözüm süresinin bu problemde daha düşük olduğu görülmüştür. Fakat bu durum 4 makineli sistem için geçerlidir. Makine sayısının artması durumunda ROHDA çözüm süresinde çok büyük bir değişiklik olmamaktadır.

Problemlerde işlem sayısının yanında, öncelik ilişkilerinin de çözüm süresi ve ortalama yüzde hataya etki ettiği söylenebilir. Öncelik ilişkilerinin çok fazla olduğu bir problemin çözüm süresi daha fazla olmakta ya da ortalama yüzde hatada artış görülmektedir. Örneğin, 1. problem ile 17. problem ele alındığında ilk problem sadece 29 işlemden oluşmaktadır ve son problemin işlem sayısı ise 297'dir. İkisinde 6 makineli sistemdeki hata payları 4.8 ve 5.53'tür. Ortalama yüzde hataya göre değerlendirme yapılacak olunursa, 14. problemde hata payının en düşük seviyede olduğu görülmüştür. En yüksek seviye ise %7,6 hata payı ile 15. problemde. Bu duruma işlem sayısındaki artıştan ziyade öncelik ilişkilerindeki ilişkinin neden olduğu söylenebilir.

Bir diğer değerlendirme makine sayılarına göre değişen ϵ ve δ değerleri için yapılmıştır. Tablo 5.5' te görüldüğü üzere, ortalama yüzde hata 5 makineli sistemde ϵ düşük, δ orta seviye değerlerini aldığı en yüksek seviyededir. Bu durum incelendiğinde GAMS çözüm süresinin ROHDA çözüm süresine göre daha hızlı olduğu söylenebilir. Ortalama yüzde hatanın en düşük seviyesi ise 4 makineli sistemdedir. Bu durumda GAMS ve ROHDA çözüm süreleri incelendiğinde ise algoritma çözüm süresinin çok değişmediği, GAMS çözüm süresinin algoritma çözüm süresine göre daha düşük olduğu söylenebilir. Fakat GAMS çözüm süresinin daha düşük olması yüzde hata payını çok fazla değiştirmemiştir. ϵ ve δ değerlerine göre makine sayısı göz ardı edilip incelenirse, 3. aralıktan türetilen ϵ değeri ile 1. aralıktan türetilen δ değerinde hata payının en düşük olduğu, buna karşılık gelen GAMS çözüm değerinin algoritma çözüm süresine göre yaklaşık 1,5 kat daha fazla olduğu söylenebilir. GAMS çözüm süresinin en çok olduğu durum 6 makineli sistemdedir. Buna karşılık gelen algoritma çözüm süresi incelendiğinde ise GAMS çözüm süresinin algoritma çözüm süresinin yaklaşık 6 katı olduğu söylenebilir. Bu değer aynı zamanda ϵ değerinin yüksek, δ değerinin ise düşük aralıktan türetildiği değerdir. Sonuç olarak ROHDA çözüm süresinin ϵ ve δ

Problem No	Makine Sayısı	Ortalama Hata %	GAMS CPU (sn)	ROHDA CPU (sn)
1	4	2,20	0,88	3,36
	5	5,29	2,09	3,60
	6	4,80	5,17	3,98
2	4	1,82	1,05	3,98
	5	4,60	2,46	4,38
	6	4,66	6,62	4,66
3	4	1,76	1,18	5,42
	5	6,92	2,49	5,72
	6	5,87	5,14	6,17
4	4	1,18	1,03	3,44
	5	5,74	2,19	3,80
	6	4,00	6,30	4,30
5	4	2,17	1,57	12,04
	5	5,90	4,25	12,24
	6	4,91	8,68	12,67
6	4	3,69	0,89	3,99
	5	4,85	1,97	4,00
	6	5,86	5,63	4,30
7	4	2,77	1,53	9,99
	5	7,05	4,79	10,35
	6	4,74	10,44	10,91
8	4	1,05	15,47	12,37
	5	5,84	49,08	12,65
	6	5,97	256,39	12,76
9	4	2,75	3,78	21,43
	5	6,46	13,88	22,80
	6	5,06	77,40	21,71
10	4	1,09	1,48	7,61
	5	4,96	2,92	8,19
	6	5,21	7,36	8,03
11	4	5,36	1,61	8,01
	5	5,81	3,20	8,55
	6	5,26	46,21	8,64
12	4	1,64	1,46	12,45
	5	6,43	3,18	13,10
	6	5,06	7,45	12,74
13	4	0,63	1,66	16,55
	5	5,18	4,26	17,60
	6	4,77	12,18	16,69
14	4	0,96	1,83	19,73
	5	6,37	4,47	21,23
	6	4,43	11,94	20,04
15	4	1,75	2,90	29,23
	5	7,06	6,98	32,39
	6	6,49	16,71	30,65
16	4	4,49	2,97	28,41
	5	7,24	7,23	31,26
	6	5,97	16,86	30,99
17	4	2,47	11,23	95,18
	5	5,83	31,87	93,01
	6	5,53	101,81	89,65

Tablo 5.4: Problem Sayısına Göre

değerlerinden bağımsızdır fakat sonuç olarak ϵ ve δ değerlerinin türetildiği aralığa bağlıdır. ϵ 'un etkisi fazla değilken asıl etkinin δ değerinden kaynaklandığı görülmektedir. Etkili olan ise ϵ ve δ 'nın kendi başlarına büyüklükleri değil, bu büyüklüklerin makinelerdeki ortalama işlem sürelerine göre büyüklüklerinin ne olduğudur. 4, 5 ve 6 makineli sistemlerde farkı etkilerin görülme sebebi budur. 4 makineli bir sistemde, δ 'nın yüksek aralıktan türetilmesi ortalama hatayı düşürürken, 6 makineli bir sistemde ise tam tersine artmaktadır. Bunun sebebi de 4 ve 6 makineli sistemlerdeki ortalama makine sürelerinin ve bir çevrimin gerektirdiği δ miktarının değişmesidir.

Makine Sayısı	ϵ	δ	Ortalama Hata %	GAMS CPU (sn)	ROHDA CPU (sn)
4	ϵ_1	δ_1	3,61	3,55	17,10
		δ_2	2,02	3,12	17,22
		δ_3	1,46	1,99	17,50
	ϵ_2	δ_1	3,49	5,07	17,12
		δ_2	2,20	3,34	17,13
		δ_2	1,31	1,95	17,47
	ϵ_3	δ_1	3,27	3,95	17,06
		δ_2	1,32	2,87	17,16
		δ_3	1,31	1,96	17,46
5	ϵ_1	δ_1	5,95	17,38	17,01
		δ_2	9,94	7,72	18,18
		δ_3	3,14	4,38	18,38
	ϵ_2	δ_1	6,35	15,27	17,23
		δ_2	8,57	6,35	18,13
		δ_3	3,10	4,49	18,48
	ϵ_3	δ_1	6,30	11,95	17,31
		δ_2	7,14	6,09	18,15
		δ_3	3,24	4,37	18,54
6	ϵ_1	δ_1	3,04	97,38	16,96
		δ_2	6,46	23,91	17,92
		δ_3	7,61	10,02	17,61
	ϵ_2	δ_1	3,02	65,80	17,24
		δ_2	4,72	14,66	17,99
		δ_3	7,55	10,00	17,60
	ϵ_3	δ_1	2,39	68,88	17,33
		δ_1	4,39	18,09	17,96
		δ_1	7,70	10,13	17,75

Tablo 5.5: ϵ ve δ değerlerine göre

F-ratio değerlerine göre ortalama hata payı, GAMS çözüm süresi ve ROHDA

çözüm süresi değişimleri Tablo 5.6' teki gibidir. F-ratio değerinin 1 olduğu, tam esnek sistemlerde hata payı, esnekliğin daha düşük olduğu aralıklara göre daha düşüktür. Çözüm zamanlarına bakıldığında ise tam esnek F-ratio değerine sahip aralıklarda çözüm süresi ROHDA için daha azdır. F-ratio'nun daha sıkı aralıklarda değer almasının da GAMS çözüm süresi daha düşüktür. Bu sebeple ROHDA'nın tam esnek sistemlerde GAMS'e göre daha başarılı olduğu söylenebilir.

Makine Sayısı	F-ratio	Hata Oranı	GAMS CPU	ROHDA CPU
4	Tam Esnek	0,81	4,25	16,26
	Daha sıkı	3,64	1,93	18,24
5	Tam Esnek	4,48	11,91	16,69
	Daha sıkı	7,46	5,42	19,17
6	Tam Esnek	4,07	55,62	16,69
	Daha sıkı	6,35	15,24	17,72

Tablo 5.6: F-ratio değerlerine göre

Genel bir değerlendirme yapılacak olursa, ROHDA özellikle makine sayısının büyük olduğu tam esnek sistemlerde GAMS'e göre daha iyi sonuçlar vermiştir. ROHDA'nın performansı ise öncelik ilişkileri, atanabilirlik, işlem sayısı ve ϵ , δ parametrelerine bağlıdır. ROHDA performansı iyi olmadığı durumlarda da ortalama %5 performansla çalışır. Bu yüzden makine sayısının çok fazla olduğu durumlarda, çevrim zamanlarının hesaplanması da zorlaşacağından GAMS'e göre ROHDA tercih sebebidir. Bu tür durumlarda hata payı olmasına rağmen çözüm alternatifi sunabilir. Fakat, performansı iyileştirmek için birtakım çalışmalar yapılabilir. Örneğin, ϵ ve δ parametrelerinin türetildiği aralık değiştirilebilir, ya da yine belirli bir aralıkta türetilen α , β ve γ parametreleri değiştirilebilir. Çözüm zamanından ödün verilerek de ROHDA iyileştirilebilir. Bu durumda çözüm zamanı artsa da makine sayısının fazla olduğu durumlarda daha etkin çalışabilir. ROHDA'nın makine işlem zamanlarını belirlemek için atamalar yaptığı bölümde de değişiklikler yapılarak tam esnek olmayan sistemlerde de etkin çalışması sağlanabilir.

6. Sonuç ve Değerlendirme

Bu çalışmada N farklı işleme sahip tek tip işlerin üretildiği ve m makineli, makine yükleme boşlatma zamanlarının ve makineler arası parça transferlerinin taşıyıcı bir robot tarafından yapıldığı bir üretim sistemindeki hat dengeleme problemi ele alınmıştır. Problemden işlemler arası atanabilirlik ve öncelik ilişkileri mevcuttur. Problemin amacı, makinelere işlemlerin atamalarını gerçekleştirerek uygun makine işlem zamanını tespit etmek ve en iyi 1-birim döngüsü ile çevrim zamanını bulmaktır. Daha önce yapılan çalışmalar, ya hat dengeleme ya da robotik hücreler üzerine olduğundan her iki konuyu da kapsayan bir problem literatürde bulunmamaktadır.

Problem detaylı bir şekilde tanımlandıktan sonra, kesin çözüm yöntemi olarak kullanılmak üzere matematiksel modeller geliştirilmiştir. Crama ve van de Klundert [19]'in ispatlamış olduğu sonuç kullanılarak sadece piramitsel permutasyonlar alternatif çözümler olarak ele alınmıştır. Her bir döngü için farklı matematiksel modeller oluşturulmuştur. Makine sayısının artması ile beraber robotun kısmi bekleme yaptığı makine sayısı artmaktadır. Bu sebeple problem daha da karmaşıklaşmaktadır. Birçok döngü için de atama problemi NP-Zor'dur. Bu sebeple parametrik bir sezgisel algoritma geliştirilmiştir. Geliştirilen sezgisel, Crama ve van de Klundert [19] algoritmasını kullanmaktadır. Geliştirilen sezgiselde, makine işlem zamanlarının dengelenmesi için bir üst limit atama yöntemi geliştirilmiştir. Bu limite ait parametrelerin hangi değerler alacağı belirlenmiş ve bu değerler doğrultusunda uygulama gerçekleştirilmiştir. İşlemlerin makinelere atanması gerçekleştirdikten sonra sezgisel, bir alt rutin olarak en iyi 1-birim döngü ve çevrim zamanını bulmak için Crama ve van de Klundert [19] algoritmasını çalıştırır. Sezgiselin kullandığı parametreleri belirlemek için

öncelikle kapsamlı bir kalibrasyon çalışması yapılmış ve kaliteli çözümleri kısa zamanda çıkartmasını sağlayacak şekilde parametre değerleri belirlenmiştir. Sezgisel yöntem performansının sonuçlarının değerlendirilmesi için, kapsamlı bir deneysel çalışma yürütülmüştür. Problemin büyüklüğüne ve karmaşıklığına, dolayısıyla da sezgiselin performansına etki edeceği düşünülen çeşitli parametreler belirlenmiş belirlenen makine sayıları için ayrı ayrı denemeler yapılmıştır. Bilgisayarda kodlanan sezgisel yöntem, GAMS 12.2 kullanılarak çözdürülen optimizasyon modeli sonuçları ile karşılaştırılmıştır. Her bir makine sayısı için 1530'ar deneme yapılmıştır ve yapılan denemeler farklı parametrelere göre değerlendirilmiştir.

Sonuç olarak, geliştirilen sezgisel algoritmanın m makineli bir sistemdeki problem için hem çözüm kalitesi hem de çözüm süresi bakımından oldukça iyi sonuçlar verdiği görülmüştür. Makine sayısı büyüdükçe, optimizasyon modeli çözüm süresi artarken, algoritma çözüm süresinde ciddi bir artış gözlemlenmemektedir. Ayrıca öncelik ve atanabilirlik ilişkilerinin fazla olması, işlem sayısının çokluğu ve ardışık iki makine arası robot hareket zamanı ve robot yükleme boşaltma zamanı gibi kriterler göz önüne alındığında kısa sürede çözüm alınabilmesi için sezgisel yöntemle baş vurulmalıdır. Bu şekilde bir yaklaşımla, literatürde daha önce hiç çalışma yapılmamış olan bu alandaki boşluk doldurulmuştur.

Bu çalışmanın devamı niteliğinde olabilecek çeşitli çalışmalar yapılabilir. Bunlardan biri, makineler arası stok alanı bulundurulmasıdır. Sistemde malzeme taşımayı robot gerçekleştirdiğinden ara stok alanlarının kullanımı ile problem daha da karmaşıklaşacaktır. Bunun sonucunda da bu çalışmada kullanılan geliştirilen yöntemler doğrudan kullanılamayacak, yeni probleme adapte edilmesi ve geliştirilmesi gerekecektir. Tek tutuculu robot kullanımının, çift tutuculu ya da çift kollu robot kullanımı ile değiştirilmesi sonucunda döngü zamanlarında ciddi bir artış ve karmaşıklık gözlemlenebilir. Bu sebeple, kullanılan robot şeklinin değiştirilmesi de ele alınabilecek diğer bir problemdir. Bu durumda, hesaplanan çevrim zamanları değişeceğinden sistemin daha etkin kullanılabilmesi için bu problemde geliştirilen optimizasyon ve sezgisel modellerin çözümleri yeterli olmayacaktır. Bir başka çalışma konusu da tek tip parça üretimi yerine değişik tip parçaların üretildiği sistemlerde ortaya çıkan problemlerin ele alınmasıdır. Bu durumda hem işlemlerin makinelere atanması, hem robot hareket döngüsünün bulunması ve hem de parça sıralamasının belirlenmesi problemleri ortaya çıkacağından karmaşıklık artacaktır. Ele alınabilecek bir diğer önemli konu ise sistemde

birden fazla robotun olması durumudur. Bu problemde, her robotun aktivite çizelgesi ayrı ayrı belirlenirken robotların birbiriyle çarpışmasının da engellenmesi gerekecektir.

Bu tezde makine sayısı verildiğinde en düşük çevrim zamanını belirleme problemi ele alınmıştır. Fakat çevrim zamanı verildiğinde optimal makine sayısını bulma problemi de geçerli ve ele alınması gereken problemlerden bir tanesidir. Robotik hücrenin yapısıyla ilgili değişik alternatifler de ele alınabilir. Örneğin, robotik hücre çizelgeleme literatüründe ele alınan yerleşimlerden bir tanesi giriş ve çıkış stoklarının aynı yerde olduğu ve makinelerin robotun etrafında dizildiği, robotun raylar üzerinde doğrusal hareketler yapmak yerine sabit bir merkezde rotasyonel hareketler yaptığı hücrelerdir. Bu tip yerleşim alternatiflerinin ele alınması da faydalı olacaktır. Bir diğer önemli olan ise montaj hattı dengeleme problemlerinde de ele alındığı üzere U-tipi hatların ele alınmasıdır. Bu problemlerde de robot U yerleşiminin her iki tarafındaki makinelere de hizmet edebilecektir ve bu makineler arasındaki hareket zamanı doğrusal bir hatta göre çok daha kısa sürecektir.

Kaynakça

- [1] Ağpak, K., Gökçen, H., Assembly line balancing: Two resource constrained cases. *Int J. Production Economics*, 96:129-140, 2005.
- [2] Agrawal, P.K., The related activity concept in assembly line balancing. *International Journal of Production Research*, **23** (2):403-421, 1985.
- [3] Aktürk, M.S., Gültekin, H. and Karasan, O.E., Robotic cell scheduling with operational flexibility. *Discrete Applied Mathematics*, **145** (3):334-348, 2005.
- [4] Allahverdi, A., Al-Anzi, F.S., A PSO and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. *Computers & Operations Research*, 33:1056-1080, 2006.
- [5] Arcus, A.L., A computer method of sequencing operations for assembly lines. *International Journal of Production Research*, **4** (4):259-277, 1965.
- [6] Batur, D. P., Karasan, O. E., Akturk, M. S., Multiple part type scheduling in flexible robotic cells. *International Journal Production Economics*, 135:726-740, 2012.
- [7] Bautista, J., Pereira, J., A dynamic programming based heuristic for the assembly line balancing problem. *European Journal of Operational Research*, 194:787-794, 2009.
- [8] Bautista, J., Pereira, J., Ant Algorithms for Assembly Line Balancing. *Lecture Notes in Computer Science*, 2463:65-75, 2002.
- [9] Bautista, J., Pereira, J., Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research*, 177:2016-2032, 2007.

- [10] Baybars, İ., A survey of exact algorithms for the simple assembly line balancing. *Management Science* **32** (8):909-932, 1986.
- [11] Baybars, I., A survey of inexact algorithms for the simple assembly line balancing problem. *GSIA*, 86:82-83, 1984.
- [12] Becker, C., Scholl, A., A Survey on Problems and Methods in Generalized Assembly Line Balancing. *European Journal of Operations Research*, 168:694-715, 2006.
- [13] Brauner, N., Identical part production in robotic cells: concept, overview and open questions. *Discrete Applied Mathematics*, 156:2480-2492, 2008.
- [14] Boysen, N., Fliedner, M., A versatile algorithm for assembly line balancing. *European Journal of Operational Research*, 184:39-56, 2008.
- [15] Brauner, N., Finke, G., On cycles and permutations in robotic cells. *Mathematical and Computer Modelling*, 34:565-591, 2001.
- [16] Brauner, N., Finke, G., On the conjecture in robotic cells:new simplified proof for the three-machine cases. *INFOR*, 37:20-36, 1999.
- [17] Capacho, L., Pastor, F., Guschinskaya, O., Dolgui, A., Heuristic methods to solve the alternative subgraphs assembly line balancing problem. *International Conference on Automation Science and Engineering*, Shanghai, China, October 7-10, 2006.
- [18] Crama, Y., Kats, Y., van de Klundert, J. and Levner, E., Cyclic scheduling in robotic flowshops. *Annals of Operations Research*, 96:97-124, 2000.
- [19] Crama, Y., van de Klundert, J., Cyclic scheduling of identical parts in robotic cells. *Operations Research*, 45: 952-95, 1997.
- [20] Crama, Y., van de Klundert, J., Cyclic scheduling in 3-machine robotic flowshops. *Journal of Scheduling*, 4: 35-54, 1999.
- [21] Dar-el (Mansoor), E.M., Solving large single assembly line balancing problems-a comparative study. *AIEE Transactions*, **7** (3):302-310, 1973.
- [22] Dawande, M., Sriskandarajah, C. and Sethi, S.P., On throughput maximization in constant travel-time robotic cells. *Manufacturing and Service Operations Management*, **4** (4):296-312, 2002.

- [23] Dominici, P., Martini, A., Tiacci, L., Saetta, S., Development of a Tabu Search Algorithm for the Assembly Line Balancing Problem. *Proceeding of the Summer Simulation MultiConference*, San Jose, CA, 135-142, 2004.
- [24] Drobouchevitch, I.G., Sethi, S., Sidney, J., Sriskandarajah, C., Scheduling dual gripper robotic cell: One-unit cycles. *European Journal of Operational Research*, 171:598-631, 2006.
- [25] Elsayed, E. A., Boucher, T. O., Analisis and control of production systems. Prentice Hall International Series in Industrial and Systems Engineering, New Jersey, 1994.
- [26] Fathian, M., Kamalabadi, N., Heydari, M., Farughi, H., A petri net model for part sequencing and robot move sequence in a two machine robotic cell. *Journal of Software Engineering and Applications*, 4:603-608, 2011.
- [27] Fleszar, K., Hindi, K.S., An enumerative heuristic and reduction methods for the assembly line balancing problem. *European Journal of Operational Research*, 145:606-620, 2003.
- [28] Geismar, H.N., Dawande, M., Sethi, S.P., Sriskandarajah, C., Sequencing and scheduling in robotic cells: recent development. *Journal of Scheduling*, 8:387-426, 2005.
- [29] Geismar, H.N., Chan, L.M.A., Dawande, M., Sriskandarajah, C., Approximations to optimal k-unit cycles for single gripper and dual gripper robotic cells. *Productions and Operations Management* **17** (5):551-563, 2008.
- [30] Geismar, H.N., Dawande, M., Sethi, S.P., Sriskandarajah, C., A note on productivity gains in flexible robotic cells. *International Journal of Flexible Manufacturing Systems*, **17** (1):5-21, 2005.
- [31] Geismar, N., Manoj, U.V., Sethi, A., Sriskandarajah, C., Scheduling robotic cells served by a dual arm robot. *IIE Transactions*, **44** (3):230-248, 2012.
- [32] Gökçen, H., Erel, E., Binary integer formulation for mixed model assembly line balancing problem. *Computers & Industrial Engineering*, **34** (2):451-461, 1998.

- [33] Gültekin, H., Aktürk, M.S. and Karasan, O.E., Cyclic Scheduling of a 2 machine robotic cell with tooling constraints. *European Journal Operational Research*, 174:777-796, 2006.
- [34] Gültekin, H., Aktürk, M.S. and Karasan, O.E., Robotic cell scheduling in a three-machine robotic flexible manufacturing cell. *Computers & Operations Research*, 34:2463-2477, 2007.
- [35] Gültekin, H., Aktürk, M.S., and Karasan, O.E., Scheduling in robotic cells: process flexibility and cell layout. *International Journal of Production Research*, **46** (8):2105-2121, 2008.
- [36] Gültekin, H., Karasan, O.E. and Aktürk, M.S., Pure cycles in flexible robotic cells. *Computers and Operations Research*, 36:329-343, 2009.
- [37] Gürsoy, A., Nuriyev, U., A mathematical model for assembly line balancing problem with lean manufacturing. *The Third International Conference "Problems of Cybernetics and Informatics"*, Baku, Azerbaijan, September 6-8, 2010.
- [38] Hackman, S.T., Magazine, M.J., Wee, T.S., Fast, effective algorithms for simple assembly line balancing problems. *Operations Research*, **37** (6):916-924, 1989.
- [39] Hall, N.G., Kamoun, H., Sriskandarajah, C., Scheduling in robotic cells: classification, two and three machine cells. *Operations Research*, **45** (3):421-439, 1997.
- [40] Helgeson, W.B., Birnie, D.P., Assembly line balancing using ranked positional weight technique. *Journal of Industrial Engineering*, 12:394-398, 1961.
- [41] Hoffman, T.R., Assembly line balancing with a precedence matrix. *Management Science*, **9** (4):551-562, 1963.
- [42] Jackson, J.R., A computing procedure for a line balancing problem. *Management Science*, **2** (3):261-271, 1956.
- [43] Jin, W., Wu, S.D., A new heuristic method for mixed model assembly line balancing problem. *Computers & Industrial Engineering*, 44:159-169, 2002.

- [44] Kamalabadi, N., Gholami, S., Mirzaei, A.H., A new solution for the cyclic multiple part type three machine robotic cell problem based on the particle swarm meta-heuristic. *Journal of Industrial and Systems Engineering*, **4** (1):304-317, 2008.
- [45] Kamoun, H., Hall, N.G., Sriskandarajah, C., Scheduling in robotic cells: heuristics and cell design. *Operations Research*, **47** (6):821-835, 1999.
- [46] Kilbridge, M. D., Wester, L., A heuristic method of assembly line balancing. *Journal of Industrial Engineering*, **12** (4):292-298, 1961.
- [47] Kim, Y.K., Kim, Y., Kim, Y.J., Two-sided assembly line balancing: a genetic algorithm approach. *Production Planning & Control*, **11** (1):44-53.
- [48] Lee, T., A review of scheduling theory and methods for semiconductor manufacturing cluster tools. *Proceedings of the 2008 Winter Conference*, 2008.
- [49] Levitin, G., Rubinovitz, J., Shnits, B., A genetic algorithm for robotic assembly line balancing. *European Journal of Operational Research*, 168:811-825, 2006.
- [50] Malakooti, B., Kumar, A., A knowledge-based system for solving multi-objective assembly line balancing problems. *International Journal of Production Research*, **34** (9):2533-2552, 1996.
- [51] Martino, L., Pastor, R., Heuristic procedures for solving the general assembly line balancing problem with setups. *International Journal of Production Research*, **48** (6):1787-1804, 2010.
- [52] Miralles, C., Garcia-Sabater, J.P., Andres, C., Cardos, M., Branch and bound procedures for solving the assembly line worker assignment and balancing problem: application to sheltered work centres for disabled. *Displied Applied Mathematics*, 156:352-367, 2008.
- [53] Moodie, C. L., Young, H. H., A heuristic method of assembly line balancing for assumptions of constant or variable work element times. *Journal of Industrial Engineering*, **16** (1):23-29, 1965.

- [54] Nicosia, G.,Pacciarelli, D., Pacifici, A., Optimally balancing assembly lines with different workstations. *Displied Applied Mathematics*, 118:99-113, 2002.
- [55] Özcan, U., Çerçioğlu, H., Gökçen, H., Toklu, B., A Tabu Search Algorithm for the Parallel Assembly Line Balancing Problem. *G.U. Journal of Science*, **22** (4):313-23, 2009.
- [56] Özcan, U., Toklu, B., A tabu search algorithm for two-sided assembly line balancing. *The International Journal of Advanced Manufacturing Technology*, 43:822-829, 2009.
- [57] Park, K., Park, S., Kim, W., A heuristic for assembly line balancing problem with incompatibility, range and partial precedence constraints. *Computers & Industrial Engineering*, **32** (2):321-332, 1997.
- [58] Pastor, R., Andres, C., Duran, A., Perez, M., Tabu search algorithms for an industrial multi-product and multi-objective assembly line balancing problem, with reduction of the task dispersion. *Journal of the Operational Research Society*, 53:1317-1323, 2002.
- [59] Pastor, R., Ferrer, L., An improved mathematical program to solve the simple assembly line balancing problem. *International Journal of Production Research*, **41** (11):2943-2959, 2008.
- [60] Peeters, M., Degraeve, Z., A linear programming based lower bound for the simple assembly line balancing problem. *European Journal of Operational Research*, 168:716-731, 2006.
- [61] Raouf, A., Tsui, C.L., Elsayed, E.A., A new heuristic approach to assembly line balancing. *Computers & Industrial Engineering*, **4** (3):223-234, 1980.
- [62] Ritt, M., Costa, A.M., A comparison of formulations for the simple assembly line balancing problem. 2011.
- [63] Ruiz-Torres, A., Ho., J., Ablanedo-Rosas, H., Makespan and Workstation Utilization Minimization in a Flowshop with Operations Flexibility. *Omega*, 39:273-282, 2011.
- [64] Sabuncuoglu, İ., Erel, E., Alp, A., Ant colony optimization for the single model U-type assembly line balancing problem. *International Journal Production Economics*, 120:287-300, 2009.

- [65] Salveson, M.E., The assembly line balancing problem. *J. Industrial Engineering*, 6:18-25, 1955.
- [66] Scholl, A., Becker, C., State-of-The-Art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing, *European Journal of Operational Research*, 168:666-693, 2006.
- [67] Scholl, A., Boysen, N., Fliedner, M., The sequence-dependent assembly line balancing problem. *Operations Research Spectrum*, 30:579-609, 2008.
- [68] Scholl, A. Fliedner, M., Boysen, N., ABSALOM: Balancing assembly lines with assignment restrictions. *European Journal of Operational Research*, 200:688-701, 2010.
- [69] Scholl, A., Klein, R., SALOME: A bidirectional branch and bound procedure for assembly line balancing. *Inform J. Computing*, 9:319-334, 1997.
- [70] Sethi, A., Sethi, S., Flexibility in Manufacturing: A Survey. *The International Journal of Flexible Manufacturing Systems*, 2:289-328, 1990.
- [71] Sethi, S.P., Sidney, J.B., Sriskandarajah, C., Scheduling in dual gripper robotic cells for productivity gains. *IEEE Transactions on Robotics and Automation*, **17** (3):324-341, 2001.
- [72] Sethi, S.P., Sriskandarajah, C., Sorger, G., Blazewicz, J., Kubiak, W., Sequencing of parts and robot moves in robotic cells. *The International Journal of Flexible Manufacturing Systems*, 4: 331-358, 1992.
- [73] Sriskandarajah, C., Drobouchevitch, I.G., Sethi, S., Chandrasekaran, R., Scheduling multiple parts in a robotic cell served by a dual gripper robot. *Operations Research*, **52** (1):65-82, 2004.
- [74] Sriskandarajah, C., Hall, N.G., Kamoun, H., Scheduling large robotic cells without buffers. *Annals of Operations Research*, 76:287-321, 1998.
- [75] Tonge, F.M., A heuristic program for assembly line balancing. *Prentice-Hall*, Englewood Cliffs, NJ., 1961.
- [76] Vilarinho, P. M., Simaria, A. S., ANTBAL: an ant colony optimization algorithm for balancing mixed-model assembly lines with parallel workstations. *International Journal of Production Research*, **44** (2):291-303, 2006.

[77] www.assembly-line-balancing.de

[78] Xiaofeng, H., Erfei, W., Ye, J., A station-oriented enumerative algorithm for two-sided assembly line balancing. *European Journal of Operational Research*, 186:435-440, 2008.

EKLER

A. 4, 5 ve 6 Makineli Robotik Hücreler İçin Çevrim Zamanları

4 makineli sistem için döngü zamanları:

4 makineli sistemde $2^{4-1} = 8$ adet piramitsel permutasyon bulunmaktadır. Bunların herbirine ait çevrim zamanları aşağıdaki gibidir:

1. $A_0A_4A_3A_2A_1: T_1^4 = 10\epsilon + 16\delta + w_1 + w_2 + w_3 + w_4$

$$w_1 = \max\{0, p_1 - 6\epsilon - 12\delta - w_2 - w_3 - w_4\}$$

$$w_2 = \max\{0, p_2 - 6\epsilon - 12\delta - w_3 - w_4\}$$

$$w_3 = \max\{0, p_3 - 6\epsilon - 12\delta - w_1 - w_4\}$$

$$w_4 = \max\{0, p_4 - 6\epsilon - 12\delta - w_1 - w_2\}$$

2. $A_0A_2A_3A_4A_1: T_2^4 = 10\epsilon + 12\delta + p_3 + p_4 + w_1 + w_2$

$$w_1 = \max\{0, p_1 - 6\epsilon - 8\delta - p_3 - p_4 - w_2\}$$

$$w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$$

3. $A_0A_3A_4A_2A_1: T_3^4 = 10\epsilon + 14\delta + p_4 + w_1 + w_2 + w_3$

$$w_1 = \max\{0, p_1 - 6\epsilon - 10\delta - p_4 - w_2 - w_3\}$$

$$w_2 = \max\{0, p_2 - 6\epsilon - 10\delta - p_4 - w_3\}$$

$$w_3 = \max\{0, p_3 - 4\epsilon - 8\delta - w_1\}$$

4. $A_0A_2A_4A_3A_1: T_4^4 = 10\epsilon + 14\delta + w_1 + w_2 + w_3 + w_4$

$$w_1 = \max\{0, p_1 - 6\epsilon - 10\delta - w_2 - w_3 - w_4\}$$

$$w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$$

$$w_3 = \max\{0, p_3 - 2\epsilon - 4\delta - w_4\}$$

$$w_4 = \max\{0, p_4 - 6\epsilon - 10\delta - w_1 - w_2\}$$

$$5. A_0A_1A_3A_4A_2: T_5^4 = 10\epsilon + 12\delta + p_1 + p_4 + w_2 + w_3$$

$$w_2 = \max\{0, p_2 - 4\epsilon - 6\delta - p_4 - w_3\}$$

$$w_3 = \max\{0, p_3 - 4\epsilon - 6\delta - p_1\}$$

$$6. A_0A_1A_4A_3A_2: T_6^4 = 10\epsilon + 14\delta + p_1 + w_2 + w_3 + w_4$$

$$w_2 = \max\{0, p_2 - 4\epsilon - 8\delta - w_3 - w_4\}$$

$$w_3 = \max\{0, p_3 - 6\epsilon - 10\delta - p_1 - w_4\}$$

$$w_4 = \max\{0, p_4 - 6\epsilon - 10\delta - p_1 - w_2\}$$

$$7. A_0A_1A_2A_4A_3: T_7^4 = 10\epsilon + 12\delta + p_1 + p_2 + w_3 + w_4$$

$$w_3 = \max\{0, p_3 - 2\epsilon - 4\delta - w_4\}$$

$$w_4 = \max\{0, p_4 - 6\epsilon - 8\delta - p_1 - p_2\}$$

$$8. A_0A_1A_2A_3A_4: T_8^4 = 10\epsilon + 10\delta + p_1 + p_2 + p_3 + p_4$$

5 makineli sistem için döngü zamanları:

5 makineli sistemde $2^{5-1} = 16$ adet piramitsel permutasyon bulunmaktadır.

Bunların herbirine ait çevrim zamanları aşağıdaki gibidir:

$$1. A_0A_1A_2A_3A_4A_5: T_1^5 = 12\epsilon + 12\delta + p_1 + p_2 + p_3 + p_4 + p_5$$

$$2. A_0A_2A_3A_4A_5A_1: T_2^5 = 12\epsilon + 14\delta + p_3 + p_4 + p_5 + w_1 + w_2$$

$$w_1 = \max\{0, p_1 - 8\epsilon - 10\delta - p_3 - p_4 - p_5 - w_2\}$$

$$w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$$

$$\begin{aligned}
3. \quad & A_0A_3A_4A_5A_2A_1: T_3^5 = 12\epsilon + 16\delta + p_4 + p_5 + w_1 + w_2 + w_3 \\
& w_1 = \max\{0, p_1 - 8\epsilon - 12\delta - p_4 - p_5 - w_2 - w_3\} \\
& w_2 = \max\{0, p_2 - 8\epsilon - 12\delta - p_4 - p_5 - w_3\} \\
& w_3 = \max\{0, p_3 - 4\epsilon - 8\delta - w_1\}
\end{aligned}$$

$$\begin{aligned}
4. \quad & A_0A_4A_5A_3A_2A_1: T_4^5 = 12\epsilon + 18\delta + p_5 + w_1 + w_2 + w_3 + w_4 \\
& w_1 = \max\{0, p_1 - 8\epsilon - 14\delta - p_5 - w_2 - w_3 - w_4\} \\
& w_2 = \max\{0, p_2 - 8\epsilon - 14\delta - w_3 - w_4 - p_5\} \\
& w_3 = \max\{0, p_3 - 8\epsilon - 14\delta - w_1 - w_4 - p_5\} \\
& w_4 = \max\{0, p_4 - 6\epsilon - 12\delta - w_1 - w_2\}
\end{aligned}$$

$$\begin{aligned}
5. \quad & A_0A_2A_3A_5A_4A_1: T_5^5 = 12\epsilon + 16\delta + p_3 + w_1 + w_2 + w_4 + w_5 \\
& w_1 = \max\{0, p_1 - 8\epsilon - 12\delta - p_3 - w_2 - w_4 - w_5\} \\
& w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\} \\
& w_4 = \max\{0, p_4 - 2\epsilon - 4\delta - w_5\} \\
& w_5 = \max\{0, p_5 - 8\epsilon - 12\delta - p_3 - w_1 - w_2\}
\end{aligned}$$

$$\begin{aligned}
6. \quad & A_0A_3A_5A_4A_2A_1: T_6^5 = 12\epsilon + 18\delta + w_1 + w_2 + w_3 + w_4 + w_5 \\
& w_1 = \max\{0, p_1 - 8\epsilon - 14\delta - w_2 - w_3 - w_4 - w_5\} \\
& w_2 = \max\{0, p_2 - 8\epsilon - 14\delta - w_3 - w_4 - w_5\} \\
& w_3 = \max\{0, p_3 - 4\epsilon - 8\delta - w_1\} \\
& w_4 = \max\{0, p_4 - 2\epsilon - 4\delta - w_5\} \\
& w_5 = \max\{0, p_5 - 8\epsilon - 14\delta - w_1 - w_2 - w_3\}
\end{aligned}$$

$$\begin{aligned}
7. \quad & A_0A_2A_4A_5A_3A_1: T_7^5 = 12\epsilon + 16\delta + p_5 + w_1 + w_2 + w_3 + w_4 \\
& w_1 = \max\{0, p_1 - 8\epsilon - 12\delta - p_5 - w_2 - w_3 - w_4\} \\
& w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\} \\
& w_3 = \max\{0, p_3 - 4\epsilon - 6\delta - p_5 - w_4\} \\
& w_4 = \max\{0, p_4 - 6\epsilon - 10\delta - w_1 - w_2\}
\end{aligned}$$

$$\begin{aligned}
8. \quad & A_0A_2A_5A_4A_3A_1: T_8^5 = 12\epsilon + 18\delta + w_1 + w_2 + w_3 + w_4 + w_5 \\
& w_1 = \max\{0, p_1 - 8\epsilon - 14\delta - w_2 - w_3 - w_4 - w_5\} \\
& w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}
\end{aligned}$$

$$\begin{aligned}
w_3 &= \max\{0, p_3 - 4\epsilon - 8\delta - p_5 - w_4\} \\
w_4 &= \max\{0, p_4 - 8\epsilon - 14\delta - w_1 - w_2 - w_5\} \\
w_5 &= \max\{0, p_5 - 8\epsilon - 14\delta - w_1 - w_2 - w_3\}
\end{aligned}$$

$$9. A_0A_5A_4A_3A_2A_1: T_9^5 = 12\epsilon + 20\delta + w_1 + w_2 + w_3 + w_4 + w_5$$

$$\begin{aligned}
w_1 &= \max\{0, p_1 - 8\epsilon - 16\delta - w_2 - w_3 - w_4 - w_5\} \\
w_2 &= \max\{0, p_2 - 8\epsilon - 16\delta - w_1 - w_3 - w_4 - w_5\} \\
w_3 &= \max\{0, p_3 - 8\epsilon - 16\delta - w_1 - w_4 - w_5\} \\
w_4 &= \max\{0, p_4 - 8\epsilon - 16\delta - w_1 - w_2 - w_5\} \\
w_5 &= \max\{0, p_5 - 8\epsilon - 16\delta - w_1 - w_2 - w_3\}
\end{aligned}$$

$$10. A_0A_1A_3A_4A_5A_2: T_{10}^5 = 12\epsilon + 14\delta + p_1 + p_4 + p_5 + w_2 + w_3$$

$$\begin{aligned}
w_2 &= \max\{0, p_2 - 6\epsilon - 8\delta - p_4 - p_5 - w_3\} \\
w_3 &= \max\{0, p_3 - 4\epsilon - 6\delta - p_1\}
\end{aligned}$$

$$11. A_0A_1A_4A_5A_3A_2: T_{11}^5 = 12\epsilon + 16\delta + p_1 + p_5 + w_2 + w_3 + w_4$$

$$\begin{aligned}
w_2 &= \max\{0, p_2 - 6\epsilon - 10\delta - p_5 - w_3 - w_4\} \\
w_3 &= \max\{0, p_3 - 8\epsilon - 12\delta - p_1 - w_4\} \\
w_4 &= \max\{0, p_4 - 6\epsilon - 10\delta - p_1 - w_2\}
\end{aligned}$$

$$12. A_0A_1A_5A_4A_3A_2: T_{12}^5 = 12\epsilon + 16\delta + p_1 + w_2 + w_3 + w_4 + w_5$$

$$\begin{aligned}
w_2 &= \max\{0, p_2 - 6\epsilon - 12\delta - w_3 - w_4 - w_5\} \\
w_3 &= \max\{0, p_3 - 8\epsilon - 14\delta - p_1 - w_4 - w_5\} \\
w_4 &= \max\{0, p_4 - 8\epsilon - 14\delta - p_1 - w_2 - w_5\} \\
w_5 &= \max\{0, p_5 - 8\epsilon - 14\delta - p_1 - w_2 - w_3\}
\end{aligned}$$

$$13. A_0A_1A_3A_5A_4A_2: T_{13}^5 = 12\epsilon + 16\delta + p_1 + w_2 + w_3 + w_4 + w_5$$

$$\begin{aligned}
w_2 &= \max\{0, p_2 - 6\epsilon - 10\delta - w_3 - w_4 - w_5\} \\
w_3 &= \max\{0, p_3 - 4\epsilon - 6\delta - p_1\} \\
w_4 &= \max\{0, p_4 - 2\epsilon - 4\delta - w_5\} \\
w_5 &= \max\{0, p_5 - 8\epsilon - 12\delta - p_1 - w_2 - w_3\}
\end{aligned}$$

$$14. A_0A_1A_2A_4A_5A_3: T_{14}^5 = 12\epsilon + 14\delta + p_1 + p_2 + p_5 + w_3 + w_4$$

$$w_3 = \max\{0, p_3 - 4\epsilon - 6\delta - p_5 - w_4\}$$

$$w_4 = \max\{0, p_4 - 6\epsilon - 8\delta - p_1 - p_2\}$$

$$15. A_0A_1A_2A_5A_4A_3: T_{15}^5 = 12\epsilon + 16\delta + p_1 + p_2 + w_3 + w_4 + w_5$$

$$w_3 = \max\{0, p_3 - 4\epsilon - 8\delta - w_4 - w_5\}$$

$$w_4 = \max\{0, p_4 - 8\epsilon - 12\delta - p_1 - p_2 - w_5\}$$

$$w_5 = \max\{0, p_5 - 8\epsilon - 10\delta - p_1 - p_2 - w_3\}$$

$$16. A_0A_1A_2A_3A_5A_4: T_{16}^5 = 12\epsilon + 14\delta + p_1 + p_2 + p_3 + w_4 + w_5$$

$$w_4 = \max\{0, p_4 - 2\epsilon - 4\delta - w_5\}$$

$$w_5 = \max\{0, p_5 - 8\epsilon - 8\delta - p_1 - p_2 - p_3\}$$

6 makineli sistem için döngü zamanları:

6 makineli sistemde $2^{6-1} = 32$ adet piramitsel permutasyon bulunmaktadır.

Bunların herbirine ait çevrim zamanları aşağıdaki gibidir:

$$1. A_0A_2A_3A_4A_5A_6A_1: T_1^6 = 14\epsilon + 16\delta + p_3 + p_4 + p_5 + p_6 + w_1 + w_2$$

$$w_1 = \max\{0, p_1 - 10\epsilon - 12\delta - p_3 - p_4 - p_5 - p_6 - w_2\}$$

$$w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$$

$$2. A_0A_3A_4A_5A_6A_2A_1: T_2^6 = 14\epsilon + 18\delta + p_4 + p_5 + p_6 + w_1 + w_2 + w_3$$

$$w_1 = \max\{0, p_1 - 10\epsilon - 14\delta - p_4 - p_5 - p_6 - w_2 - w_3\}$$

$$w_2 = \max\{0, p_2 - 10\epsilon - 14\delta - p_4 - p_5 - p_6 - w_3\}$$

$$w_3 = \max\{0, p_3 - 4\epsilon - 8\delta - w_1\}$$

$$3. A_0A_4A_5A_6A_3A_2A_1: T_3^6 = 14\epsilon + 20\delta + p_5 + p_6 + w_1 + w_2 + w_3 + w_4$$

$$w_1 = \max\{0, p_1 - 10\epsilon - 16\delta - p_5 - p_6 - w_2 - w_3 - w_4\}$$

$$w_2 = \max\{0, p_2 - 10\epsilon - 16\delta - p_5 - p_6 - w_3 - w_4\}$$

$$w_3 = \max\{0, p_3 - 10\epsilon - 16\delta - p_5 - p_6 - w_1 - w_4\}$$

$$w_4 = \max\{0, p_4 - 6\epsilon - 12\delta - w_1 - w_2\}$$

$$4. A_0A_5A_6A_4A_3A_2A_1: T_4^6 = 14\epsilon + 22\delta + p_6 + w_1 + w_2 + w_3 + w_4 + w_5$$

$$w_1 = \max\{0, p_1 - 10\epsilon - 18\delta - p_6 - w_2 - w_3 - w_4 - w_5\}$$

$$w_2 = \max\{0, p_2 - 10\epsilon - 18\delta - p_6 - w_3 - w_4 - w_5\}$$

$$w_3 = \max\{0, p_3 - 10\epsilon - 16\delta - p_6 - w_1 - w_4 - w_5\}$$

$$w_4 = \max\{0, p_4 - 10\epsilon - 18\delta - p_6 - w_1 - w_2 - w_5\}$$

$$w_5 = \max\{0, p_5 - 8\epsilon - 16\delta - w_1 - w_2 - w_3\}$$

$$5. A_0A_2A_3A_5A_6A_4A_1: T_5^6 = 14\epsilon + 18\delta + p_3 + p_6 + w_1 + w_2 + w_4 + w_5$$

$$w_1 = \max\{0, p_1 - 10\epsilon - 14\delta - p_3 - p_6 - w_2 - w_4 - w_5\}$$

$$w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$$

$$w_4 = \max\{0, p_4 - 4\epsilon - 6\delta - p_6 - w_5\}$$

$$w_5 = \max\{0, p_5 - 8\epsilon - 12\delta - p_3 - w_1 - w_2\}$$

$$6. A_0A_3A_5A_6A_4A_2A_1: T_6^6 = 14\epsilon + 20\delta + p_6 + w_1 + w_2 + w_3 + w_4 + w_5$$

$$w_1 = \max\{0, p_1 - 10\epsilon - 16\delta - p_6 - w_2 - w_3 - w_4 - w_5\}$$

$$w_2 = \max\{0, p_2 - 10\epsilon - 16\delta - p_6 - w_3 - w_4 - w_5\}$$

$$w_3 = \max\{0, p_3 - 4\epsilon - 8\delta - w_1\}$$

$$w_4 = \max\{0, p_4 - 4\epsilon - 8\delta - p_6 - w_5\}$$

$$w_5 = \max\{0, p_5 - 8\epsilon - 14\delta - w_1 - w_2 - w_3\}$$

$$7. A_0A_3A_4A_6A_5A_2A_1: T_7^6 = 14\epsilon + 20\delta + p_4 + w_1 + w_2 + w_3 + w_5 + w_6$$

$$w_1 = \max\{0, p_1 - 10\epsilon - 16\delta - p_4 - w_2 - w_3 - w_5 - w_6\}$$

$$w_2 = \max\{0, p_2 - 10\epsilon - 16\delta - p_4 - w_3 - w_5 - w_6\}$$

$$w_3 = \max\{0, p_3 - 4\epsilon - 8\delta - w_1\}$$

$$w_5 = \max\{0, p_5 - 2\epsilon - 4\delta - w_6\}$$

$$w_6 = \max\{0, p_6 - 10\epsilon - 16\delta - p_4 - w_1 - w_2 - w_3\}$$

$$8. A_0A_3A_6A_5A_4A_2A_1: T_8^6 = 14\epsilon + 22\delta + w_1 + w_2 + w_3 + w_4 + w_5 + w_6$$

$$w_1 = \max\{0, p_1 - 10\epsilon - 18\delta - w_2 - w_3 - w_4 - w_5 - w_6\}$$

$$w_2 = \max\{0, p_2 - 10\epsilon - 18\delta - w_3 - w_4 - w_5 - w_6\}$$

$$w_3 = \max\{0, p_3 - 4\epsilon - 8\delta - w_1\}$$

$$\begin{aligned}
w_4 &= \max\{0, p_4 - 4\epsilon - 8\delta - w_5 - w_6\} \\
w_5 &= \max\{0, p_5 - 10\epsilon - 18\delta - w_1 - w_2 - w_3 - w_6\} \\
w_6 &= \max\{0, p_6 - 10\epsilon - 18\delta - w_1 - w_2 - w_3 - w_4\}
\end{aligned}$$

$$9. A_0A_4A_6A_5A_3A_2A_1: T_9^6 = 14\epsilon + 22\delta + w_1 + w_2 + w_3 + w_4 + w_5 + w_6$$

$$\begin{aligned}
w_1 &= \max\{0, p_1 - 10\epsilon - 18\delta - w_2 - w_3 - w_4 - w_5 - w_6\} \\
w_2 &= \max\{0, p_2 - 10\epsilon - 14\delta - w_3 - w_4 - w_5 - w_6\} \\
w_3 &= \max\{0, p_3 - 10\epsilon - 18\delta - w_1 - w_4 - w_5 - w_6\} \\
w_4 &= \max\{0, p_4 - 6\epsilon - 12\delta - w_1 - w_2\} \\
w_5 &= \max\{0, p_5 - 2\epsilon - 4\delta - w_6\} \\
w_6 &= \max\{0, p_6 - 10\epsilon - 18\delta - w_1 - w_2 - w_3 - w_4\}
\end{aligned}$$

$$10. A_0A_2A_4A_5A_6A_3A_1: T_{10}^6 = 14\epsilon + 18\delta + p_5 + p_6 + w_1 + w_2 + w_3 + w_4$$

$$\begin{aligned}
w_1 &= \max\{0, p_1 - 10\epsilon - 14\delta - p_5 - p_6 - w_2 - w_3 - w_4\} \\
w_2 &= \max\{0, p_2 - 2\epsilon - 4\delta\} \\
w_3 &= \max\{0, p_3 - 6\epsilon - 8\delta - p_5 - p_6 - w_4\} \\
w_4 &= \max\{0, p_4 - 6\epsilon - 10\delta - w_1 - w_2\}
\end{aligned}$$

$$11. A_0A_6A_5A_4A_3A_2A_1: T_{11}^6 = 14\epsilon + 24\delta + w_1 + w_2 + w_3 + w_4 + w_5 + w_6$$

$$\begin{aligned}
w_1 &= \max\{0, p_1 - 10\epsilon - 20\delta - w_2 - w_3 - w_4 - w_5 - w_6\} \\
w_2 &= \max\{0, p_2 - 10\epsilon - 20\delta - w_3 - w_4 - w_5 - w_6\} \\
w_3 &= \max\{0, p_3 - 10\epsilon - 20\delta - w_1 - w_4 - w_5 - w_6\} \\
w_4 &= \max\{0, p_4 - 10\epsilon - 20\delta - w_1 - w_2 - w_5 - w_6\} \\
w_5 &= \max\{0, p_5 - 10\epsilon - 20\delta - w_1 - w_2 - w_3 - w_6\} \\
w_6 &= \max\{0, p_6 - 10\epsilon - 20\delta - w_1 - w_2 - w_3 - w_4\}
\end{aligned}$$

$$12. A_0A_2A_5A_6A_4A_3A_1: T_{12}^6 = 14\epsilon + 20\delta + p_6 + w_1 + w_2 + w_3 + w_4 + w_5$$

$$\begin{aligned}
w_1 &= \max\{0, p_1 - 10\epsilon - 16\delta - p_6 - w_2 - w_3 - w_4 - w_5\} \\
w_2 &= \max\{0, p_2 - 2\epsilon - 4\delta\} \\
w_3 &= \max\{0, p_3 - 6\epsilon - 10\delta - p_6 - w_4 - w_5\} \\
w_4 &= \max\{0, p_4 - 10\epsilon - 16\delta - p_6 - w_1 - w_2 - w_5\} \\
w_5 &= \max\{0, p_5 - 8\epsilon - 14\delta - w_1 - w_2 - w_3\}
\end{aligned}$$

13. $A_0A_2A_6A_5A_4A_3A_1: T_{13}^6 = 14\epsilon + 22\delta + w_1 + w_2 + w_3 + w_4 + w_5 + w_6$
 $w_1 = \max\{0, p_1 - 10\epsilon - 18\delta - w_1 - w_2 - w_3 - w_4 - w_5 - w_6\}$
 $w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$
 $w_3 = \max\{0, p_3 - 6\epsilon - 12\delta - w_4 - w_5 - w_6\}$
 $w_4 = \max\{0, p_4 - 10\epsilon - 18\delta - w_1 - w_2 - w_5 - w_6\}$
 $w_5 = \max\{0, p_5 - 10\epsilon - 18\delta - w_1 - w_2 - w_3 - w_6\}$
 $w_6 = \max\{0, p_6 - 10\epsilon - 18\delta - w_1 - w_2 - w_3 - w_4\}$
14. $A_0A_2A_3A_6A_5A_4A_1: T_{14}^6 = 14\epsilon + 20\delta + p_3 + w_1 + w_2 + w_4 + w_5 + w_6$
 $w_1 = \max\{0, p_1 - 10\epsilon - 16\delta - p_3 - w_2 - w_4 - w_5 - w_6\}$
 $w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$
 $w_4 = \max\{0, p_4 - 4\epsilon - 8\delta - w_5 - w_6\}$
 $w_5 = \max\{0, p_5 - 10\epsilon - 16\delta - w_1 - w_2 - w_3 - w_6\}$
 $w_6 = \max\{0, p_6 - 10\epsilon - 16\delta - p_3 - w_1 - w_2 - w_4\}$
15. $A_0A_2A_3A_4A_6A_5A_1: T_{15}^6 = 14\epsilon + 18\delta + p_3 + p_4 + w_1 + w_2 + w_5 + w_6$
 $w_1 = \max\{0, p_1 - 10\epsilon - 14\delta - p_3 - p_4 - w_2 - w_5 - w_6\}$
 $w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$
 $w_5 = \max\{0, p_5 - 2\epsilon - 4\delta - w_6\}$
 $w_6 = \max\{0, p_6 - 10\epsilon - 14\delta - p_3 - p_4 - w_1 - w_2\}$
16. $A_0A_2A_4A_6A_5A_3A_1: T_{16}^6 = 14\epsilon + 20\delta + w_1 + w_2 + w_3 + w_4 + w_5 + w_6$
 $w_1 = \max\{0, p_1 - 10\epsilon - 16\delta - w_2 - w_3 - w_4 - w_5 - w_6\}$
 $w_2 = \max\{0, p_2 - 2\epsilon - 4\delta\}$
 $w_3 = \max\{0, p_3 - 6\epsilon - 10\delta - w_4 - w_5 - w_6\}$
 $w_4 = \max\{0, p_4 - 6\epsilon - 10\delta - w_1 - w_2\}$
 $w_5 = \max\{0, p_5 - 2\epsilon - 4\delta - w_6\}$
 $w_6 = \max\{0, p_6 - 10\epsilon - 16\delta - w_1 - w_2 - w_3 - w_4\}$
17. $A_0A_1A_3A_4A_5A_6A_2: T_{17}^6 = 14\epsilon + 16\delta + p_1 + p_4 + p_5 + p_6 + w_2 + w_3$
 $w_2 = \max\{0, p_2 - 8\epsilon - 10\delta - p_4 - p_5 - p_6 - w_3\}$
 $w_3 = \max\{0, p_3 - 4\epsilon - 6\delta - p_1\}$

18. $A_0A_1A_4A_5A_6A_3A_2: T_{18}^6 = 14\epsilon + 18\delta + p_1 + p_5 + p_6 + w_2 + w_3 + w_4$
 $w_2 = \max\{0, p_2 - 8\epsilon - 12\delta - p_5 - p_6 - w_3 - w_4\}$
 $w_3 = \max\{0, p_3 - 10\epsilon - 14\delta - p_1 - p_5 - p_6 - w_4\}$
 $w_4 = \max\{0, p_4 - 6\epsilon - 10\delta - p_1 - w_2\}$
19. $A_0A_1A_5A_6A_4A_3A_2: T_{19}^6 = 14\epsilon + 20\delta + p_1 + p_6 + w_2 + w_3 + w_4 + w_5$
 $w_2 = \max\{0, p_2 - 8\epsilon - 14\delta - p_6 - w_3 - w_4 - w_5\}$
 $w_3 = \max\{0, p_3 - 10\epsilon - 16\delta - p_1 - p_6 - w_4 - w_5\}$
 $w_4 = \max\{0, p_4 - 10\epsilon - 16\delta - p_1 - p_6 - w_2 - w_5\}$
 $w_5 = \max\{0, p_5 - 8\epsilon - 14\delta - p_1 - w_2 - w_3\}$
20. $A_0A_1A_6A_5A_4A_3A_2: T_{20}^6 = 14\epsilon + 22\delta + p_1 + w_2 + w_3 + w_4 + w_5 + w_6$
 $w_2 = \max\{0, p_2 - 8\epsilon - 16\delta - w_3 - w_4 - w_5 - w_6\}$
 $w_3 = \max\{0, p_3 - 10\epsilon - 18\delta - p_1 - w_4 - w_5 - w_6\}$
 $w_4 = \max\{0, p_4 - 10\epsilon - 18\delta - p_1 - w_2 - w_5 - w_6\}$
 $w_5 = \max\{0, p_5 - 10\epsilon - 18\delta - p_1 - w_2 - w_3 - w_6\}$
 $w_6 = \max\{0, p_6 - 10\epsilon - 18\delta - p_1 - w_2 - w_3 - w_4\}$
21. $A_0A_1A_3A_5A_6A_4A_2: T_{21}^6 = 14\epsilon + 18\delta + p_1 + p_6 + w_2 + w_3 + w_4 + w_5$
 $w_2 = \max\{0, p_2 - 8\epsilon - 12\delta - p_6 - w_3 - w_4 - w_5\}$
 $w_3 = \max\{0, p_3 - 4\epsilon - 6\delta - p_1\}$
 $w_4 = \max\{0, p_4 - 4\epsilon - 6\delta - p_6 - w_5\}$
 $w_5 = \max\{0, p_5 - 8\epsilon - 12\delta - p_1 - w_2 - w_3\}$
22. $A_0A_1A_4A_6A_5A_3A_2: T_{22}^6 = 14\epsilon + 20\delta + p_1 + w_2 + w_3 + w_4 + w_5 + w_6$
 $w_2 = \max\{0, p_2 - 8\epsilon - 14\delta - w_3 - w_4 - w_5 - w_6\}$
 $w_3 = \max\{0, p_3 - 10\epsilon - 16\delta - p_1 - w_4 - w_5 - w_6\}$
 $w_4 = \max\{0, p_4 - 6\epsilon - 10\delta - p_1 - w_2\}$
 $w_5 = \max\{0, p_5 - 2\epsilon - 4\delta - w_6\}$
 $w_6 = \max\{0, p_6 - 10\epsilon - 16\delta - p_1 - w_2 - w_3 - w_4\}$
23. $A_0A_1A_3A_4A_6A_5A_2: T_{23}^6 = 14\epsilon + 18\delta + p_1 + p_4 + w_2 + w_3 + w_5 + w_6$
 $w_2 = \max\{0, p_2 - 8\epsilon - 12\delta - p_4 - w_3 - w_5 - w_6\}$

$$\begin{aligned}
w_3 &= \max\{0, p_3 - 4\epsilon - 6\delta - p_1\} \\
w_5 &= \max\{0, p_5 - 2\epsilon - 4\delta - w_6\} \\
w_6 &= \max\{0, p_6 - 10\epsilon - 14\delta - p_1 - p_4 - w_2 - w_3\}
\end{aligned}$$

$$24. A_0A_1A_3A_6A_5A_4A_2: T_{24}^6 = 14\epsilon + 20\delta + p_1 + w_2 + w_3 + w_4 + w_5 + w_6$$

$$\begin{aligned}
w_2 &= \max\{0, p_2 - 8\epsilon - 14\delta - w_3 - w_4 - w_5 - w_6\} \\
w_3 &= \max\{0, p_3 - 4\epsilon - 6\delta - p_1\} \\
w_4 &= \max\{0, p_4 - 4\epsilon - 8\delta - w_5 - w_6\} \\
w_5 &= \max\{0, p_5 - 10\epsilon - 16\delta - p_1 - w_2 - w_3 - w_6\} \\
w_6 &= \max\{0, p_6 - 10\epsilon - 16\delta - p_1 - w_2 - w_3 - w_4\}
\end{aligned}$$

$$25. A_0A_1A_2A_4A_5A_6A_3: T_{25}^6 = 14\epsilon + 16\delta + p_1 + p_2 + p_5 + p_6 + w_3 + w_4$$

$$\begin{aligned}
w_3 &= \max\{0, p_3 - 6\epsilon - 8\delta - p_5 - p_6 - w_4\} \\
w_4 &= \max\{0, p_4 - 6\epsilon - 8\delta - p_1 - p_2\}
\end{aligned}$$

$$26. A_0A_1A_2A_6A_5A_4A_3: T_{26}^6 = 14\epsilon + 20\delta + p_1 + P_2 + w_3 + w_4 + w_5 + w_6$$

$$\begin{aligned}
w_3 &= \max\{0, p_3 - 6\epsilon - 12\delta - w_4 - w_5 - w_6\} \\
w_4 &= \max\{0, p_4 - 10\epsilon - 16\delta - p_1 - p_2 - w_5 - w_6\} \\
w_5 &= \max\{0, p_5 - 10\epsilon - 16\delta - p_1 - p_2 - w_3 - w_6\} \\
w_6 &= \max\{0, p_6 - 10\epsilon - 16\delta - p_1 - p_2 - w_3 - w_4\}
\end{aligned}$$

$$27. A_0A_1A_2A_4A_6A_5A_3: T_{27}^6 = 14\epsilon + 18\delta + p_1 + p_2 + w_3 + w_4 + w_5 + w_6$$

$$\begin{aligned}
w_3 &= \max\{0, p_3 - 6\epsilon - 10\delta - w_4 - w_5 - w_6\} \\
w_4 &= \max\{0, p_4 - 6\epsilon - 8\delta - p_1 - p_2\} \\
w_5 &= \max\{0, p_5 - 2\epsilon - 4\delta - w_6\} \\
w_6 &= \max\{0, p_6 - 10\epsilon - 14\delta - p_1 - p_2 - w_3 - w_4\}
\end{aligned}$$

$$28. A_0A_1A_2A_5A_6A_4A_3: T_{28}^6 = 14\epsilon + 18\delta + p_1 + p_2 + p_6 + w_3 + w_4 + w_5$$

$$\begin{aligned}
w_3 &= \max\{0, p_3 - 6\epsilon - 10\delta - p_1 - p_2\} \\
w_4 &= \max\{0, p_4 - 10\epsilon - 14\delta - p_1 - p_2 - p_6 - w_5\} \\
w_5 &= \max\{0, p_5 - 8\epsilon - 12\delta - p_1 - p_2 - w_3\}
\end{aligned}$$

29. $A_0A_1A_2A_3A_5A_6A_4$: $T_{29}^6 = 14\epsilon + 16\delta + p_1 + p_2 + p_3 + p_6 + w_4 + w_5$
 $w_4 = \max\{0, p_4 - 4\epsilon - 6\delta - p_6 - w_5\}$
 $w_5 = \max\{0, p_5 - 8\epsilon - 10\delta - p_1 - p_2 - p_3\}$
30. $A_0A_1A_2A_3A_6A_5A_4$: $T_{30}^6 = 14\epsilon + 18\delta + p_1 + p_2 + p_3 + w_4 + w_5 + w_6$
 $w_4 = \max\{0, p_4 - 4\epsilon - 8\delta - w_5 - w_6\}$
 $w_5 = \max\{0, p_5 - 10\epsilon - 14\delta - p_1 - p_2 - p_3\}$
 $w_6 = \max\{0, p_6 - 10\epsilon - 14\delta - p_1 - p_2 - p_3 - w_4\}$
31. $A_0A_1A_2A_3A_4A_6A_5$: $T_{31}^6 = 14\epsilon + 16\delta + p_1 + p_2 + p_3 + p_4 + w_5 + w_6$
 $w_5 = \max\{0, p_5 - 2\epsilon - 4\delta - w_6\}$
 $w_6 = \max\{0, p_6 - 10\epsilon - 12\delta - p_1 - p_2 - p_3 - p_4\}$
32. $A_0A_1A_2A_3A_4A_6A_5$: $T_{32}^6 = 14\epsilon + 14\delta + p_1 + p_2 + p_3 + p_4 + p_5 + p_6$

B. Deneysel Çalışma Sonuçları

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
1	1	1	90	1,51	93	3,43	3,33
		2	108,4	1,40	110,6	3,29	2,06
		3	191,2	1,19	191,2	3,48	0,00
	2	1	94	1,35	97	3,40	3,19
		2	114,8	1,41	116,6	3,30	1,64
		3	192,8	1,08	192,8	3,48	0,00
	3	1	108,4	1,40	110,6	3,29	2,06
		2	126	1,14	126,2	3,30	0,17
		3	198	1,00	198,8	3,53	0,50
2	1	1	142,6	1,32	145,6	4,03	2,10
		2	183,2	1,41	184,4	3,99	0,77
		3	257,8	1,17	259	4,08	0,51
	2	1	145	1,40	148	4,01	2,07
		2	183,2	1,35	184,4	3,96	0,75
		3	274,4	1,21	275,2	4,09	0,33
	3	1	158,6	1,27	161,6	3,99	1,89
		2	184,4	1,15	185,6	4,01	0,73
		3	264,8	1,33	265,2	4,07	0,16
3	1	1	161,2	1,43	166,2	5,42	3,10
		2	222,4	1,27	222,4	5,40	0,00
		3	278,8	1,12	282,4	5,47	1,32
	2	1	167,6	1,30	172,6	5,40	2,99
		2	189,6	1,36	193,6	5,34	2,17
		3	311,6	1,03	313,2	5,67	0,60
	3	1	180,4	1,37	185,4	5,41	2,78
		2	219,6	1,24	222,6	5,38	1,55
		3	321,6	1,11	321,6	5,51	0,00
4	1	1	4329,2	1,58	4391,2	3,49	1,43
		2	5792,8	1,41	5807,2	3,49	0,30
		3	7792,4	1,35	7834,8	3,60	0,61
	2	1	4578,8	1,53	4640,8	3,46	1,35
		2	5772	1,51	5784,4	3,49	0,26
		3	8119,6	1,25	8149,6	3,61	0,42
	3	1	4709,2	1,44	4771,2	3,46	1,32
		2	6159,6	1,45	6191,2	3,49	0,52
		3	8442,8	1,25	8459,6	3,61	0,23
5	1	1	142,8	1,92	143,8	11,91	0,70
		2	193,2	1,93	193,4	12,19	0,12
		3	257	1,82	259,2	12,44	0,94
	2	1	146	1,80	147	11,84	0,69
		2	166,8	1,92	167,6	11,79	0,50
		3	279	1,86	280	12,68	0,41
	3	1	164,4	1,84	165,4	11,79	0,61
		2	196	1,88	196,4	11,96	0,23
		3	273,6	1,81	274,8	12,34	0,47
6	1	1	91,4	1,27	95,4	4,17	4,38
		2	110,2	1,19	112,8	4,05	2,49
		3	178,4	1,07	179,2	4,18	0,50
	2	1	94,6	1,23	98,6	4,24	4,23
		2	119,2	1,10	120,8	4,06	1,50
		3	172,4	0,99	174	4,15	0,99
	3	1	100,2	1,29	104,2	4,20	3,99
		2	119,4	1,18	121,8	4,07	2,23
		3	180,4	1,09	182	4,16	0,98

Tablo B.1: 4 makine $F - ratio = 0$ için ayrıntılı sonuçlar (1-6)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
7	1	1	1071,6	2,54	1077,6	9,49	0,56
		2	1366	1,97	1367,2	9,78	0,10
		3	2035,2	1,57	2039,2	10,07	0,23
	2	1	1117,2	2,14	1123,2	9,48	0,54
		2	1404,4	1,94	1405,6	9,60	0,10
		3	2158	1,58	2158	10,07	0,00
	3	1	1179,6	2,33	1185,6	9,45	0,51
		2	1529,8	1,96	1530,6	9,67	0,05
		3	2127,6	1,65	2127,6	10,03	0,00
8	1	1	23366,2	28,12	23601,2	12,40	1,01
		2	28482	26,87	28576	12,67	0,38
		3	43360	5,15	43590,6	12,88	0,61
	2	1	23911	82,58	24146	12,37	0,98
		2	31823,8	23,02	31903,8	12,64	0,27
		3	43782,6	3,21	44115,6	12,85	0,87
	3	1	25603	40,83	25838	12,34	0,92
		2	31356,2	19,79	31403,2	12,60	0,17
		3	44873,8	3,32	45010,8	13,11	0,35
9	1	1	46589,6	7,32	46626,6	19,19	0,08
		2	60349,6	5,85	60357	19,71	0,01
		3	88407	2,55	88844,6	19,93	0,59
	2	1	47664	8,41	47701	19,15	0,08
		2	59076	6,08	59090,8	19,54	0,03
		3	91461	2,84	91851,8	20,16	0,52
	3	1	52110,4	7,98	52147,4	19,13	0,07
		2	67045	5,55	67392	20,03	0,45
		3	98908,4	2,60	98908,4	20,08	0,00
10	1	1	4452,2	2,40	4564,2	6,81	2,52
		2	5169	2,23	5236,2	6,78	1,39
		3	7951,4	1,78	8017,6	7,09	0,96
	2	1	4638,6	2,37	4750,6	6,78	2,42
		2	5508,4	2,11	5553,2	6,83	0,91
		3	8278,6	1,74	8322,8	7,07	0,62
	3	1	4914,6	2,47	5026,6	6,73	2,28
		2	5988,4	1,97	6029	6,91	0,70
		3	8113,2	1,96	8165,2	7,08	0,72
11	1	1	470,2	2,72	484,2	7,12	2,98
		2	583,6	2,82	589,2	6,96	1,10
		3	911,4	1,76	916,4	7,34	0,64
	2	1	479,8	2,84	493,8	6,94	2,92
		2	580,4	2,23	586	6,96	1,09
		3	828,8	1,80	835,4	7,13	0,85
	3	1	523,8	2,76	537,8	6,91	2,68
		2	669,4	2,25	672,2	7,03	0,52
		3	902	1,71	904,8	7,18	0,35
12	1	1	447	1,68	453	10,71	1,34
		2	523,4	1,77	527	10,70	0,74
		3	819	1,35	825,8	11,01	0,93
	2	1	471	1,62	477	10,66	1,27
		2	561,6	1,70	564	10,74	0,48
		3	856,2	1,36	859,8	11,07	0,48
	3	1	501,4	1,67	507,4	10,64	1,20
		2	616,4	1,73	618,8	10,81	0,47
		3	895,6	1,34	899	11,20	0,45

Tablo B.2: 4 makine $F - ratio = 0$ için ayrıntılı sonuçlar (7-12)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
13	1	1	500,6	2,28	500,6	14,14	0,00
		2	621,8	2,19	621,8	14,21	0,00
		3	926,6	1,92	931	14,67	0,54
	2	1	523	2,34	523	14,02	0,00
		2	608,4	2,16	608,4	14,10	0,00
		3	993,6	1,90	998,2	14,79	0,57
	3	1	545,4	2,29	545,4	13,99	0,00
		2	736	2,14	736	14,49	0,00
		3	979,2	1,87	982,8	14,62	0,42
14	1	1	1259,2	2,45	1260,2	17,17	0,08
		2	1682,6	2,13	1685,4	17,35	0,15
		3	2460,8	1,85	2472	17,62	0,54
	2	1	1325,6	2,49	1326,6	17,07	0,08
		2	1620,4	2,29	1620,8	17,29	0,03
		3	2503,8	1,83	2514	17,66	0,48
	3	1	1404	2,58	1405	17,11	0,07
		2	1724,4	2,34	1724,6	17,38	0,01
		3	2567,4	1,84	2574,8	17,77	0,33
15	1	1	1735,4	3,54	1744,4	27,84	0,52
		2	2030,6	3,38	2036	28,06	0,29
		3	3339,6	2,87	3343,8	29,00	0,16
	2	1	1812,2	3,55	1821,2	27,84	0,50
		2	2204,8	3,35	2208,4	28,14	0,19
		3	3459,2	2,70	3459,2	29,38	0,00
	3	1	1890,6	3,49	1899,6	27,84	0,48
		2	2218,4	3,41	2222	28,34	0,18
		3	3716,4	2,70	3716,4	29,18	0,00
16	1	1	1329	3,70	1333	27,40	0,30
		2	1753,6	3,39	1755,2	28,43	0,11
		3	2865,2	2,76	2865,2	28,67	0,00
	2	1	1421	3,48	1425	27,28	0,28
		2	1642,6	3,25	1643,4	27,43	0,05
		3	2413,8	3,11	2448,6	28,27	1,57
	3	1	2634,4	2,76	2634,4	28,48	0,00
		2	1520,2	3,64	1524,2	27,25	0,26
		3	2030	3,14	2031,6	28,31	0,07
17	1	1	20502,6	16,83	20571,6	86,47	0,34
		2	29422,6	13,09	29460,8	89,50	0,12
		3	42294,8	9,31	42294,8	90,44	0,00
	2	1	21613,8	17,02	21682,8	88,10	0,32
		2	27960,4	15,32	28011	88,24	0,19
		3	40359,4	9,95	40537,4	91,00	0,51
	3	1	23582,6	19,70	23651,6	87,08	0,29
		2	26559,2	13,32	26559,2	87,46	0,00
		3	43442	10,36	43730,8	91,18	0,80

Tablo B.3: 4 makine $F - ratio = 0$ için ayrıntılı sonuçlar (13-17)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
1	1	1	105,20	0,53	110,80	3,57	5,82
		2	125,60	0,42	130,80	3,30	4,46
		3	150,20	0,34	150,20	3,20	0,00
	2	1	109,20	0,54	114,80	3,54	5,59
		2	132,00	0,43	136,00	3,26	3,26
		3	159,00	0,40	159,00	3,20	0,00
	3	1	119,60	0,57	125,20	3,46	5,13
		2	143,80	0,45	145,00	3,23	0,91
		3	169,40	0,47	170,40	3,19	0,62
2	1	1	174,00	0,96	180,20	4,04	3,98
		2	196,40	0,83	202,60	3,93	3,58
		3	259,80	0,77	265,00	3,93	1,93
	2	1	181,20	0,81	187,40	3,97	3,86
		2	205,80	0,78	209,40	3,89	2,12
		3	279,20	0,70	283,40	3,92	1,49
	3	1	188,40	0,88	194,60	3,94	3,71
		2	205,60	0,86	208,60	3,92	1,62
		3	295,80	0,73	299,00	3,93	1,04
3	1	1	206,40	1,15	212,20	5,38	2,83
		2	225,60	1,13	231,40	5,37	2,58
		3	314,20	1,04	317,40	5,37	0,94
	2	1	207,20	1,23	213,00	5,38	2,81
		2	230,20	1,12	234,60	5,37	1,97
		3	316,20	1,01	317,40	5,31	0,41
	3	1	222,40	1,13	228,20	5,41	2,60
		2	249,40	1,11	254,60	5,41	2,08
		3	306,20	1,08	309,40	5,50	0,94
4	1	1	4717,60	0,70	4803,20	3,42	1,83
		2	5682,00	0,67	5718,40	3,30	0,63
		3	9630,80	0,48	9909,60	3,42	2,84
	2	1	4930,40	0,78	5016,00	3,39	1,74
		2	5974,80	0,60	5995,60	3,27	0,32
		3	9188,00	0,54	9464,80	3,39	2,81
	3	1	5226,40	0,80	5312,00	3,36	1,65
		2	6351,60	0,65	6371,60	3,29	0,29
		3	9416,40	0,55	9683,60	3,39	2,68
5	1	1	168,00	1,31	175,80	12,15	4,72
		2	199,20	1,29	205,80	11,84	3,58
		3	288,80	1,24	299,80	12,08	4,29
	2	1	170,40	1,34	178,20	11,92	4,77
		2	188,80	1,26	196,60	11,83	4,18
		3	296,80	1,19	302,80	12,21	2,22
	3	1	184,80	1,29	192,60	11,83	4,38
		2	205,60	1,38	209,40	11,87	1,91
		3	274,40	1,19	286,60	12,08	4,35
6	1	1	110,40	0,77	119,80	3,95	8,91
		2	122,40	0,69	131,80	3,87	7,82
		3	182,40	0,51	184,20	3,79	1,06
	2	1	114,40	0,66	123,80	3,94	8,58
		2	133,20	0,59	139,80	3,81	4,98
		3	169,60	0,69	171,00	3,81	0,74
	3	1	120,00	0,63	129,40	3,90	8,11
		2	144,00	0,61	148,00	3,75	2,84
		3	184,00	0,54	187,40	3,77	2,05

Tablo B.4: 4 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (1-6)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
7	1	1	1148,60	1,15	1251,60	10,39	8,64
		2	1445,80	1,18	1512,00	10,29	4,84
		3	2194,60	0,93	2221,80	10,16	1,28
	2	1	1191,80	1,19	1294,80	10,38	8,46
		2	1436,00	1,10	1521,80	10,12	6,25
		3	2123,60	1,02	2168,00	10,05	2,08
	3	1	1267,00	1,14	1370,00	10,51	7,94
		2	1558,40	1,12	1634,60	10,11	5,24
		3	2211,80	1,00	2277,20	10,23	3,03
8	1	1	24683,60	7,55	25112,20	11,94	1,85
		2	30466,20	3,21	30514,60	11,91	0,14
		3	41324,20	1,78	42369,20	12,31	2,80
	2	1	25513,20	5,39	25941,80	11,99	1,80
		2	29454,20	13,57	29860,40	12,05	1,54
		3	46378,40	1,35	46553,00	12,42	0,45
	3	1	27284,40	6,88	27713,00	11,97	1,68
		2	32267,60	3,91	32283,40	12,06	0,05
		3	44506,40	2,00	45689,80	12,23	2,99
9	1	1	54560,40	2,63	59728,80	23,30	9,67
		2	69117,40	2,15	70686,60	23,16	2,20
		3	91433,20	1,63	92437,20	23,07	1,00
	2	1	57238,00	2,45	62406,40	23,31	9,19
		2	64840,60	2,14	69634,60	23,26	8,13
		3	91696,20	1,66	93095,60	23,06	1,64
	3	1	59692,40	2,23	64860,80	23,41	8,78
		2	71931,40	2,19	75195,80	23,18	4,97
		3	93119,80	1,71	94904,40	23,01	2,01
10	1	1	5060,80	0,95	5111,00	8,33	0,94
		2	6016,00	0,82	6065,80	8,29	0,85
		3	8564,40	0,70	8667,40	8,38	1,15
	2	1	5365,60	0,96	5415,80	8,27	0,91
		2	6438,00	0,88	6456,80	8,30	0,29
		3	8372,80	0,74	8411,60	8,40	0,51
	3	1	5545,60	0,90	5595,80	8,35	0,86
		2	6294,60	0,93	6344,40	8,20	0,79
		3	8955,20	0,72	9028,60	8,37	0,77
11	1	1	551,80	1,04	632,40	9,04	16,23
		2	621,40	0,89	693,20	8,95	12,63
		3	817,40	0,92	842,00	8,89	3,06
	2	1	553,40	1,05	634,00	9,03	16,17
		2	598,20	0,97	678,80	8,94	14,82
		3	1017,40	0,74	1039,00	8,96	2,13
	3	1	621,40	0,99	702,00	8,95	14,03
		2	766,40	0,88	785,80	8,90	2,63
		3	987,80	0,68	1005,20	8,93	1,71
12	1	1	483,80	1,46	494,00	14,12	2,12
		2	610,20	1,38	614,80	13,89	0,85
		3	860,20	1,21	909,60	14,13	6,23
	2	1	515,00	1,43	525,20	14,10	1,99
		2	583,80	1,34	592,20	13,93	1,53
		3	889,40	1,26	920,60	13,99	3,67
	3	1	551,80	1,40	562,00	13,91	1,85
		2	651,40	1,32	655,20	13,89	0,62
		3	914,20	1,21	942,20	14,63	3,32

Tablo B.5: 4 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (7-12)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
13	1	1	627,40	1,33	634,20	18,73	1,11
		2	689,80	1,24	696,60	18,70	1,01
		3	891,20	1,18	895,80	18,91	0,57
	2	1	644,20	1,28	651,00	18,68	1,07
		2	758,60	1,12	763,80	19,07	0,72
		3	1003,60	1,12	1032,00	18,71	2,57
	3	1	681,80	1,20	688,60	18,72	1,03
		2	766,20	1,25	771,40	18,61	0,65
		3	1022,80	1,09	1033,00	18,67	1,10
14	1	1	1431,20	1,64	1441,00	22,10	0,69
		2	1676,20	1,52	1681,60	21,88	0,33
		3	2611,00	1,27	2731,80	22,38	4,93
	2	1	1464,80	1,57	1474,60	22,06	0,67
		2	1741,00	1,53	1745,80	22,06	0,31
		3	2484,20	1,34	2562,60	22,17	3,36
	3	1	1609,60	1,55	1619,40	22,01	0,61
		2	1769,40	1,46	1771,20	21,97	0,11
		3	2561,40	1,32	2679,00	22,18	4,52
15	1	1	1796,00	2,83	1900,00	30,01	5,98
		2	2137,60	2,74	2167,80	30,32	1,64
		3	3133,60	2,41	3216,80	30,18	2,52
	2	1	1889,60	2,72	1993,60	29,96	5,69
		2	2253,40	2,62	2265,60	29,89	0,63
		3	3188,80	2,36	3282,20	30,16	2,87
	3	1	2016,00	2,75	2120,00	29,94	5,28
		2	2600,40	2,56	2614,60	30,23	0,49
		3	3330,80	2,29	3461,40	29,91	4,19
16	1	1	1357,80	3,03	1562,00	28,99	14,96
		2	1706,80	2,71	1775,60	29,04	4,53
		3	2832,80	2,47	2959,00	28,69	4,60
	2	1	1415,40	2,90	1619,60	28,88	14,51
		2	1749,20	2,63	1858,60	28,78	6,62
		3	2460,00	2,46	2624,80	28,58	6,78
	3	1	1461,00	2,80	1665,20	28,98	14,10
		2	1740,40	2,75	1880,60	28,84	8,61
		3	2792,20	2,47	2886,20	29,01	3,43
17	1	1	26747,20	8,94	28351,20	100,77	7,83
		2	29169,60	8,87	30773,60	100,86	6,92
		3	38801,80	8,23	39329,80	104,08	1,39
	2	1	28144,00	8,79	29748,00	101,27	6,98
		2	29756,20	8,86	31360,20	101,05	6,81
		3	38874,00	8,29	39454,20	102,67	1,62
	3	1	29204,00	8,78	30808,00	100,61	7,14
		2	33570,80	8,43	34468,60	100,86	3,11
		3	42697,00	8,01	42747,80	101,55	0,12

Tablo B.6: 4 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (13-17)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
1	1	1	79,4	2,6592	81,4	3,32	2,52
		2	109,8	2,3292	122,4	3,51	11,64
		3	192	1,7024	193	3,70	0,63
	2	1	80,2	2,9358	82,2	3,32	2,49
		2	109,8	2,4944	121,6	3,51	10,54
		3	194,4	1,6636	196,2	3,70	1,15
	3	1	109,8	2,3292	122,4	3,51	11,64
		2	112,4	2,5052	122,4	3,45	8,95
		3	185,6	1,6664	188,4	3,73	1,74
2	1	1	113,8	3,6904	116,8	4,18	2,64
		2	159,4	2,789	180	4,43	12,97
		3	336	2,3148	337,6	4,50	0,67
	2	1	118,6	3,7222	121,6	4,14	2,53
		2	207,2	3,2394	228,4	4,50	10,34
		3	290,4	2,3238	292,8	4,48	1,02
	3	1	137,4	3,096	143,2	4,14	4,06
		2	184,6	3,1674	201,6	4,37	9,20
		3	303,2	2,0232	303,2	4,48	0,00
3	1	1	136,6	3,9194	138,6	5,07	1,46
		2	158	3,7934	175,2	5,30	10,87
		3	381,6	2,7986	381,6	5,79	0,00
	2	1	136,6	3,773	138,6	5,08	1,46
		2	235,2	3,3766	259,2	5,84	10,25
		3	324,8	3,0206	324,8	5,83	0,00
	3	1	158,4	3,7848	168,8	5,20	6,54
		2	200	3,5032	218,4	5,67	9,22
		3	366,4	3,1378	371,2	5,80	1,72
4	1	1	3683,6	3,2448	3906	3,49	5,96
		2	5514,4	2,924	6018,8	3,85	9,79
		3	9532,8	1,7706	9532,8	3,86	0,00
	2	1	4063,6	3,1424	4365,2	3,54	7,25
		2	6139,2	2,6618	6549,6	3,88	7,26
		3	10014,4	1,8652	10014,4	3,85	0,00
	3	1	4025,6	3,1848	4219,2	3,49	4,68
		2	5470,4	2,8528	5974,4	3,81	9,30
		3	9546,4	1,8712	9546,4	3,88	0,00
5	1	1	115,6	6,0008	118,6	11,22	2,60
		2	159,6	4,693	178,6	12,15	11,77
		3	326,4	3,6572	326,4	12,58	0,00
	2	1	120	5,4848	123,2	11,24	2,66
		2	180,8	4,4804	200,8	12,37	11,13
		3	296	3,8658	297,8	12,55	0,76
	3	1	136,2	5,4982	142	11,34	4,16
		2	201,4	4,5128	217,6	12,47	8,12
		3	301,6	3,8078	301,6	12,58	0,00
6	1	1	77	3,0356	79	3,78	2,60
		2	110,8	2,2024	125	4,09	12,85
		3	201,6	1,5518	202,2	4,25	0,38
	2	1	79,4	2,8988	81,4	3,77	2,52
		2	122	2,0874	136,2	4,18	11,70
		3	200,8	1,6498	202,2	4,26	0,90
	3	1	84,8	3,5762	87,2	3,74	2,80
		2	114,6	2,1534	126	4,03	9,95
		3	192	1,8496	194	4,29	1,21

Tablo B.7: 5 makine $F - ratio = 0$ için ayrıntılı sonuçlar (1-6)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
7	1	1	910,6	9,7376	951,2	9,14	4,22
		2	1525,6	5,0576	1648,4	10,36	8,54
		3	2658,4	3,576	2658,4	10,79	0,00
	2	1	942,6	7,9124	986,4	9,20	4,45
		2	1602,4	4,811	1719,4	10,35	7,58
		3	2636	3,5442	2636	10,78	0,00
	3	1	1023,2	6,5336	1076,8	9,35	4,94
		2	1690,4	4,5428	1801,4	10,39	6,70
		3	2634,4	3,5724	2634,4	10,82	0,00
8	1	1	19613,2	244,2346	20495	11,82	4,33
		2	28164	75,8976	31063,6	12,73	10,61
		3	50398,4	4,8958	50398,4	12,89	0,00
	2	1	21228,6	170,3056	22469,2	12,06	5,65
		2	31936	21,9368	33964,8	12,82	6,95
		3	49328	4,7804	49328	12,91	0,00
	3	1	22338,6	143,4612	23753	12,16	6,24
		2	32773,6	25,4622	34796,2	12,83	6,49
		3	48380	4,5268	48380	12,90	0,00
9	1	1	37660,4	46,912	38223,2	18,49	1,42
		2	62700	15,4896	66504,6	20,72	7,11
		3	99883,2	9,514	99883,2	21,05	0,00
	2	1	40087,8	46,5322	41119,2	18,68	2,35
		2	59311,8	17,5862	63629,6	20,51	7,98
		3	108743,2	11,6478	108743,2	21,09	0,00
	3	1	43625	40,8168	45237	18,93	3,41
		2	75485,6	10,6866	77183,8	21,04	2,45
		3	85936,8	7,7358	85936,8	21,05	0,00
10	1	1	3552,2	4,7984	3610,8	6,88	1,64
		2	5967,2	3,547	6393,4	7,41	7,71
		3	8940	2,939	8940	7,42	0,00
	2	1	3805,4	4,144	3982,8	6,94	4,46
		2	5537	3,8178	5918	7,38	6,98
		3	9655,2	2,958	9655,2	7,43	0,00
	3	1	4263,6	3,9778	4468,2	7,04	4,68
		2	5756,8	3,8846	6235,2	7,37	8,31
		3	10169,6	2,8548	10169,6	7,42	0,00
11	1	1	393	6,368	412,4	6,74	4,83
		2	622,6	3,7934	676,4	7,62	9,48
		3	1062,4	2,7512	1062,4	7,65	0,00
	2	1	413,6	7,2268	431,6	6,75	4,26
		2	637,6	3,4044	686,2	7,62	8,09
		3	1044	2,7592	1044	7,67	0,00
	3	1	441	4,8704	462,8	6,79	4,87
		2	664	3,4946	711,8	7,63	7,50
		3	1032	2,768	1032	7,69	0,00
12	1	1	381	4,624	389,6	10,09	2,27
		2	616	3,6854	690	11,75	12,01
		3	1153,6	2,7546	1153,6	11,58	0,00
	2	1	416,4	4,2308	444,4	10,53	6,45
		2	518,4	4,0288	564,4	11,20	9,80
		3	826,4	3,0174	826,4	11,67	0,00
	3	1	402,4	4,271	411	10,06	2,15
		2	549,6	3,6328	598,8	11,49	8,96
		3	956	2,8864	956	11,69	0,00

Tablo B.8: 5 makine $F - ratio = 0$ için ayrıntılı sonuçlar (7-12)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
13	1	1	435,8	5,6572	461,6	14,05	5,74
		2	646	4,7848	698,8	15,30	8,66
		3	1176,8	3,4968	1176,8	15,51	0,00
	2	1	431,2	7,2298	452	13,96	4,67
		2	682,4	4,7212	736,4	15,46	8,26
		3	1207,2	3,5942	1207,2	15,55	0,00
	3	1	476,8	6,9298	501,6	14,13	5,00
		2	732,8	4,6334	778	15,53	6,45
		3	1145,6	3,54	1145,6	15,56	0,00
14	1	1	1068	6,8238	1094,6	16,65	2,46
		2	1590,2	5,4162	1763,2	18,68	11,15
		3	2841,6	3,998	2841,6	18,97	0,00
	2	1	1180,4	6,6452	1232,6	17,01	4,17
		2	1684	5,2536	1840,4	18,73	9,62
		3	2888,8	4,0618	2888,8	19,01	0,00
	3	1	1296,6	6,6244	1380,4	17,52	6,27
		2	1708,8	5,1618	1849,6	18,66	8,41
		3	2752,8	4,0054	2752,8	18,96	0,00
15	1	1	1507	9,9934	1601,6	27,19	6,00
		2	2087,8	8,5046	2327	29,41	11,43
		3	4204,8	6,449	4204,8	30,82	0,00
	2	1	1629	8,9644	1759,2	27,76	7,87
		2	2338,4	7,8014	2545,2	29,71	9,07
		3	3823,2	6,4514	3823,2	30,87	0,00
	3	1	1733,6	9,1606	1838,4	27,92	5,88
		2	2513,6	7,8276	2690,4	29,81	7,15
		3	3828,8	6,3954	3828,8	30,89	0,00
16	1	1	1189	9,2686	1268,2	26,26	6,49
		2	1730	7,9624	1931,4	28,91	11,78
		3	3121,6	6,5456	3121,6	30,45	0,00
	2	1	1179	9,7982	1217,2	25,86	3,16
		2	1707	7,8442	1867,4	28,84	9,66
		3	2951,2	6,4218	2951,2	30,12	0,00
	3	1	3138,4	6,5774	3138,4	30,15	0,00
		2	1371,6	8,6928	1460	27,18	6,19
		3	1804,4	8,2004	1941,4	28,80	7,92
17	1	1	17539,4	76,8212	17959,4	83,78	2,24
		2	29647,2	34,865	32638	92,74	10,31
		3	47384,8	23,9172	47384,8	96,03	0,00
	2	1	19130,6	55,043	19986,2	84,83	4,30
		2	26419	38,5648	28844,4	90,79	9,43
		3	45008,8	24,136	45008,8	95,26	0,00
	3	1	20790,2	55,6494	21841,8	86,80	4,78
		2	31016	34,752	33109	91,84	6,93
		3	47131,2	23,7902	47131,2	95,65	0,00

Tablo B.9: 5 makine $F - ratio = 0$ için ayrıntılı sonuçlar (13-17)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
1	1	1	100,00	2,15	108,60	3,65	8,62
		2	132,80	2,08	138,00	3,65	4,03
		3	208,00	1,51	217,60	3,83	5,00
	2	1	101,60	2,12	110,20	3,69	8,46
		2	127,60	1,99	132,00	3,66	3,83
		3	195,60	1,55	205,60	3,86	5,19
	3	1	108,00	2,16	116,60	3,59	7,96
		2	136,80	1,79	141,00	3,63	3,47
		3	177,80	1,66	188,40	3,76	5,79
2	1	1	147,20	2,41	154,00	4,33	4,84
		2	162,80	2,31	171,60	4,22	5,93
		3	332,80	1,46	333,20	4,56	0,12
	2	1	146,00	2,40	151,60	4,40	3,96
		2	187,80	2,01	201,60	4,27	7,39
		3	311,20	1,63	320,00	4,64	3,67
	3	1	155,60	2,23	161,20	4,25	3,78
		2	221,40	1,95	236,80	4,41	7,06
		3	276,60	1,52	283,60	4,61	2,59
3	1	1	202,00	1,69	225,20	5,89	10,71
		2	233,00	1,48	258,80	5,92	10,48
		3	332,00	1,34	364,20	5,96	10,16
	2	1	206,80	1,60	230,00	5,85	10,41
		2	245,40	1,61	268,60	5,93	9,16
		3	360,00	1,42	381,60	6,06	6,50
	3	1	210,00	1,70	233,20	5,85	10,31
		2	258,20	1,48	281,20	5,93	8,23
		3	348,80	1,35	372,80	6,01	7,05
4	1	1	3903,60	2,15	4129,20	3,63	5,74
		2	5398,80	1,86	5841,60	3,86	8,24
		3	9977,20	1,36	10498,00	3,97	5,41
	2	1	4220,00	2,03	4396,40	3,63	4,21
		2	5995,60	1,82	6402,00	3,85	6,81
		3	8734,00	1,55	9432,00	4,04	8,21
	3	1	4363,60	1,99	4634,00	3,69	6,15
		2	6677,60	1,72	7106,40	3,92	6,48
		3	9328,00	1,51	10026,80	4,11	7,75
5	1	1	130,40	4,18	135,40	11,80	3,75
		2	201,20	4,00	225,20	12,84	12,00
		3	355,20	3,25	381,00	12,86	7,57
	2	1	133,60	4,16	138,60	11,74	3,63
		2	190,80	3,86	205,40	12,71	7,84
		3	312,00	3,47	340,40	12,82	9,21
	3	1	149,00	3,93	156,00	11,88	4,74
		2	172,40	4,10	182,80	12,41	6,01
		3	268,80	3,62	296,40	12,83	10,27
6	1	1	95,80	1,69	105,20	4,01	10,00
		2	120,40	1,64	124,80	3,83	3,92
		3	208,00	1,30	208,00	4,06	0,00
	2	1	99,80	1,76	109,20	4,03	9,59
		2	126,00	1,75	130,20	3,85	4,28
		3	187,20	1,35	189,80	4,08	1,60
	3	1	105,40	1,85	114,80	3,96	9,03
		2	133,40	1,68	136,40	3,80	2,54
		3	192,80	1,39	195,40	4,02	1,48

Tablo B.10: 5 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (1-6)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
7	1	1	979,40	3,64	1102,20	10,10	12,11
		2	1270,40	3,03	1458,20	10,91	15,24
		3	2249,60	2,32	2355,40	10,83	5,07
	2	1	1047,20	14,71	1181,20	10,26	12,45
		2	1444,00	2,68	1626,80	10,79	12,91
		3	2436,00	2,37	2538,20	10,56	4,60
	3	1	1141,20	3,21	1282,20	10,06	12,13
		2	1450,00	2,70	1581,00	10,63	9,61
		3	2127,20	2,32	2259,20	11,03	6,31
8	1	1	20633,20	64,77	22995,60	12,20	11,03
		2	34478,00	4,82	37251,00	12,72	7,98
		3	48860,00	2,91	51347,40	12,83	5,10
	2	1	21273,60	77,06	23478,00	12,50	10,09
		2	32806,60	7,26	35120,20	13,28	7,08
		3	49751,20	3,05	52569,40	13,04	5,83
	3	1	24418,40	17,52	26536,40	12,53	8,85
		2	32712,00	7,36	34228,20	12,84	4,83
		3	50452,40	3,14	52283,20	12,65	4,00
9	1	1	56140,20	5,39	63866,60	24,50	13,76
		2	68171,80	4,77	75142,80	25,44	10,32
		3	99457,60	3,92	106477,40	25,22	7,19
	2	1	56343,40	5,90	63717,80	25,42	12,96
		2	64925,20	5,03	72806,20	25,68	12,51
		3	99707,20	3,97	106292,60	25,24	6,77
	3	1	58774,40	5,52	65622,60	25,73	11,44
		2	71016,20	4,60	78102,20	25,79	10,18
		3	101793,60	3,89	108109,80	25,75	6,47
10	1	1	4435,20	2,60	4687,80	9,18	5,80
		2	5597,00	2,34	5939,00	9,10	6,53
		3	9146,60	1,84	9852,60	9,28	7,68
	2	1	4691,20	2,51	5008,20	9,19	7,14
		2	6690,80	2,20	7133,00	9,09	6,28
		3	10484,80	1,72	10870,00	8,89	3,62
	3	1	5195,20	2,39	5498,40	9,02	6,39
		2	6780,20	2,18	7097,00	9,19	4,56
		3	9225,80	1,96	9910,00	9,22	7,48
11	1	1	445,40	2,87	498,80	9,43	12,25
		2	720,00	2,16	776,40	10,14	7,58
		3	1109,60	1,59	1138,80	9,83	3,36
	2	1	465,20	3,13	519,40	9,47	11,95
		2	711,60	2,23	760,80	9,60	6,80
		3	1124,80	1,55	1152,00	9,79	2,38
	3	1	490,00	2,83	538,60	9,48	9,89
		2	695,20	2,04	746,20	9,86	7,43
		3	992,80	1,79	1027,80	10,09	3,85
12	1	1	395,80	3,18	415,80	14,56	4,92
		2	559,40	2,73	639,20	15,36	14,74
		3	1043,20	2,16	1121,80	15,31	7,88
	2	1	414,20	3,29	439,60	14,26	5,90
		2	606,40	2,73	681,20	15,27	12,54
		3	1076,80	2,15	1124,20	15,52	4,73
	3	1	465,00	2,88	497,00	14,78	6,82
		2	625,60	2,66	686,40	15,28	9,87
		3	1021,60	2,27	1088,40	15,48	6,68

Tablo B.11: 5 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (7-12)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
13	1	1	520,80	3,87	551,20	19,57	5,77
		2	716,60	3,59	761,40	20,41	6,34
		3	1054,80	3,26	1141,80	20,49	8,32
	2	1	511,00	3,94	540,80	20,10	5,66
		2	667,40	3,65	708,80	20,33	5,96
		3	1173,60	2,99	1251,80	19,87	7,09
	3	1	563,00	3,97	592,80	20,10	5,25
		2	733,60	3,61	753,80	20,21	2,84
		3	1152,80	3,16	1235,60	20,72	7,26
14	1	1	1202,80	4,12	1246,20	23,58	3,45
		2	1787,20	3,47	2052,20	25,04	15,17
		3	2916,80	3,20	3221,00	24,06	10,75
	2	1	1286,00	3,98	1343,40	23,89	4,46
		2	1772,00	3,64	1930,40	24,67	8,62
		3	2680,00	3,25	2950,00	24,05	10,31
	3	1	1329,20	3,95	1369,20	23,43	3,13
		2	1706,80	3,70	1817,60	24,80	6,34
		3	2685,60	3,16	2965,20	24,36	10,41
15	1	1	1489,60	7,04	1628,20	34,68	9,33
		2	2412,40	5,75	2651,40	35,68	10,38
		3	3910,40	5,41	4222,40	35,44	7,96
	2	1	1562,80	6,87	1713,80	34,80	9,74
		2	2410,00	5,63	2602,80	36,45	7,98
		3	3620,00	5,47	3963,40	35,59	9,53
	3	1	1675,00	6,85	1834,60	34,47	9,52
		2	2546,80	5,65	2750,40	35,31	8,43
		3	3764,80	5,45	4016,80	36,20	6,83
16	1	1	1207,00	7,04	1307,20	33,54	8,36
		2	1588,40	6,55	1784,80	33,29	12,55
		3	2662,80	5,91	2873,80	34,07	8,31
	2	1	1212,00	7,60	1335,40	33,35	10,66
		2	1886,40	6,23	2095,40	34,74	11,31
		3	2917,20	6,01	3106,00	34,15	6,84
	3	1	1364,40	7,25	1493,40	33,78	9,59
		2	1814,40	6,30	2007,60	34,25	11,08
		3	3126,00	6,00	3320,40	34,91	6,32
17	1	1	19368,60	24,42	21769,80	91,44	12,49
		2	25888,40	22,17	27905,60	96,70	7,88
		3	48079,20	21,49	50460,60	94,32	5,05
	2	1	19618,00	25,96	22385,20	94,43	14,07
		2	31490,80	23,41	33213,40	94,65	5,51
		3	49471,20	21,52	52169,60	99,04	5,60
	3	1	22653,60	23,20	24767,00	92,86	9,41
		2	30261,80	22,43	30842,80	95,84	1,91
		3	43961,00	21,46	46181,80	97,11	5,12

Tablo B.12: 5 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (13-17)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
1	1	1	73	10,52	74,8	3,75	2,86
		2	128,8	7,61	131,6	4,14	2,30
		3	232,4	5,55	245	3,85	6,11
	2	1	73,4	10,86	75,2	3,62	2,69
		2	138	6,85	138	4,12	0,00
		3	215,6	5,98	229,2	3,89	6,95
	3	1	128,8	7,61	131,6	4,14	2,30
		2	143,6	7,00	143,6	4,20	0,00
		3	213,2	6,03	223,6	3,95	5,38
2	1	1	120,8	10,73	121,2	4,45	0,30
		2	213	7,69	221,2	4,91	3,57
		3	369,2	5,50	392,8	4,57	6,87
	2	1	125,2	11,16	125,6	4,48	0,32
		2	206,6	8,05	216,4	4,86	4,36
		3	366	5,59	386,8	4,60	6,06
	3	1	139,6	11,45	140	4,54	0,24
		2	215,6	7,20	220,8	4,88	2,13
		3	340	5,73	364,6	4,71	7,27
3	1	1	151,8	7,83	153,4	5,99	1,04
		2	249,2	6,04	261,2	6,36	4,50
		3	385,2	4,82	415,2	6,10	7,94
	2	1	146,8	7,21	147,6	5,92	0,71
		2	249,4	6,00	259,6	6,45	3,65
		3	379,6	4,60	410	6,18	8,22
	3	1	162,2	7,15	162,8	6,05	0,32
		2	237,2	6,08	240,8	6,48	1,45
		3	412,4	4,63	439,6	6,12	6,76
4	1	1	4293,2	9,42	4298	4,42	0,11
		2	5868,8	8,01	6013,6	4,64	1,73
		3	11472,8	6,06	12142,8	4,28	5,85
	2	1	4288	10,41	4327,2	4,40	0,82
		2	6604,8	7,75	6648,8	4,65	0,65
		3	12497,6	6,04	13167,6	4,23	5,36
	3	1	4127,2	10,23	4128,4	4,30	0,03
		2	7281,6	7,17	7782,8	4,68	6,86
		3	8896,8	6,38	9570,4	4,48	7,58
5	1	1	122,2	12,19	123,6	12,24	1,37
		2	208,8	9,73	220,8	12,84	4,94
		3	338,4	7,92	364,4	12,48	7,78
	2	1	122,8	13,56	124	12,11	0,97
		2	226,2	9,50	238	12,79	4,64
		3	359,6	7,99	385,6	12,48	7,40
	3	1	130,8	13,09	131,2	12,16	0,24
		2	219,4	9,98	226	12,90	2,66
		3	358,8	7,99	384,8	12,56	7,37
6	1	1	78,4	9,36	78,4	4,10	0,00
		2	127,2	7,32	136,2	4,51	6,18
		3	241,2	5,11	249,2	4,45	3,34
	2	1	76,6	10,93	78,2	4,03	2,04
		2	122	6,65	122	4,62	0,00
		3	269,6	4,81	277,6	4,34	2,97
	3	1	73,8	10,20	74,4	3,91	0,85
		2	142	6,64	142	4,77	0,00
		3	193,2	5,81	206,8	4,54	7,39

Tablo B.13: 6 makine $F - ratio = 0$ için ayrıntılı sonuçlar (1-6)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
7	1	1	952,4	23,61	973,4	10,03	2,07
		2	1777,6	10,28	1899,8	11,45	6,19
		3	2593,8	8,57	2805,8	11,12	8,29
	2	1	947,2	23,74	960	10,00	1,41
		2	1655,2	12,29	1738	11,59	4,43
		3	2604	8,30	2805	11,16	7,84
	3	1	1115,8	16,42	1139,2	10,47	2,12
		2	1743	10,99	1824,4	11,55	4,31
		3	2608	8,54	2799	11,15	7,45
8	1	1	21569,2	1644,81	21765,2	12,58	0,79
		2	36985,6	28,68	39620	13,00	6,40
		3	53060,4	9,31	57264,4	12,76	8,02
	2	1	22886,8	1245,12	23369,6	12,80	1,93
		2	34314,8	66,26	35944,8	13,05	4,28
		3	63086,8	9,42	67254,4	12,59	6,70
	3	1	23201,6	1234,38	23655,6	12,56	1,86
		2	38154	32,47	39539,2	13,05	3,30
		3	54761,2	9,93	58571,8	12,74	7,08
9	1	1	42480,4	156,21	42900,4	21,09	0,85
		2	66464,8	36,04	69915,2	22,19	4,29
		3	122387,6	11,09	131879,4	21,74	7,78
	2	1	43079,2	216,00	43191,6	21,30	0,29
		2	70187,2	18,75	74779,2	22,26	5,90
		3	125087,6	11,58	133439,6	21,70	6,82
	3	1	44089	173,05	44713	21,12	1,57
		2	67216	207,84	69254	22,31	2,62
		3	113002,4	11,17	121156,6	21,86	7,33
10	1	1	4342,8	10,45	4342,8	8,99	0,00
		2	6858,2	8,31	7047,6	9,25	2,74
		3	10549,6	6,62	11313,4	8,83	7,24
	2	1	3805,2	12,10	3832,4	8,43	0,73
		2	5579,2	8,90	5579,2	9,30	0,00
		3	8619,2	7,13	9387,2	8,86	8,92
	3	1	4512,4	11,25	4588,4	8,72	1,82
		2	7444,8	7,95	8084,4	9,18	8,59
		3	11106	6,53	11758,2	8,81	5,87
11	1	1	408,2	382,07	411,8	9,26	0,92
		2	730	8,75	767,2	9,90	4,65
		3	1215,6	6,54	1300	9,49	7,04
	2	1	438,4	77,19	442,4	9,43	0,92
		2	687,6	9,44	708	9,97	2,64
		3	1110	6,61	1193,6	9,58	7,65
	3	1	443,6	231,42	446	9,34	0,57
		2	719,8	9,45	742,4	9,91	2,77
		3	1105,6	6,63	1188,4	9,67	7,56
12	1	1	416,4	11,36	424,8	13,77	1,91
		2	713,8	7,88	750	14,63	4,57
		3	1232,8	6,18	1326,4	14,08	7,74
	2	1	405,2	11,30	410	13,54	1,13
		2	651,2	9,07	665,2	14,61	2,05
		3	1190,8	6,42	1273,6	14,21	7,11
	3	1	445,8	10,96	453,8	13,85	1,74
		2	674,8	8,59	686,4	14,61	1,64
		3	1038	6,41	1115,2	14,41	7,63

Tablo B.14: 6 makine $F - ratio = 0$ için ayrıntılı sonuçlar (7-12)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
13	1	1	454,6	29,38	461	17,77	1,31
		2	762	10,77	802,8	18,53	4,87
		3	1300,4	8,17	1400,4	18,20	7,74
	2	1	447	38,51	447,2	17,82	0,04
		2	778,4	10,81	823,6	18,57	5,20
		3	1184,4	8,29	1279,4	18,24	8,20
	3	1	514,8	23,36	526	18,18	2,00
		2	740,2	12,24	760,6	18,47	2,46
		3	1162,8	8,33	1252,2	18,30	7,79
14	1	1	1053,2	45,30	1060	21,04	0,63
		2	1861,4	11,00	1955,6	21,88	4,53
		3	3424	8,98	3656	21,52	6,92
	2	1	1334,6	15,68	1367	21,62	2,28
		2	1778,4	11,78	1805,2	22,06	1,28
		3	3132	8,83	3365	21,74	7,53
	3	1	1345,6	15,18	1358,4	21,74	0,87
		2	1912,6	11,55	1973,6	22,18	2,87
		3	3267,6	8,93	3500,6	21,76	7,23
15	1	1	1509,2	25,96	1533,6	28,42	1,29
		2	3120	15,28	3446,4	30,69	10,47
		3	5197,8	14,58	5557	30,27	6,91
	2	1	1782,6	22,67	1842,2	29,42	3,14
		2	3104,4	15,64	3412	30,97	9,91
		3	4396,6	14,97	4723,8	30,50	7,78
	3	1	1578,4	24,82	1607,8	28,98	1,77
		2	2850,8	16,21	3066,4	30,87	7,07
		3	4198,8	15,09	4497,2	30,88	7,33
16	1	1	1202,8	24,04	1202,8	30,81	0,00
		2	2253,6	16,14	2457,4	32,67	8,43
		3	2877,2	15,12	3133,2	32,61	8,95
	2	1	1229,6	24,98	1246,8	31,34	1,51
		2	2129,2	17,88	2228,4	32,97	4,34
		3	3558,8	15,02	3810,4	32,40	7,21
	3	1	3066	15,17	3316,4	32,58	8,23
		2	1352,4	23,04	1382	31,28	2,10
		3	1803,4	18,84	1808,8	32,51	0,25
17	1	1	16810,2	553,30	16857,4	78,72	0,27
		2	34854,8	71,65	37963,6	85,29	8,31
		3	50979,2	56,43	54965,6	86,12	7,82
	2	1	23917	138,13	24470,2	82,64	2,35
		2	28570,4	96,97	28570,4	87,21	0,00
		3	62338,8	55,38	66152	85,33	6,12
	3	1	21785,6	232,03	21785,6	85,29	0,00
		2	38354	58,27	41701,2	85,75	8,72
		3	49618,4	57,23	53638	86,18	8,11

Tablo B.15: 6 makine $F - ratio = 0$ için ayrıntılı sonuçlar (13-17)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
1	1	1	81,20	3,41	88,00	3,89	9,57
		2	140,80	2,35	150,80	4,40	6,78
		3	241,20	1,75	258,20	3,95	7,53
	2	1	77,80	3,50	84,20	3,82	8,78
		2	126,80	2,60	133,60	4,35	5,23
		3	255,20	1,82	270,40	3,88	6,23
	3	1	90,20	3,30	93,20	3,81	3,38
		2	135,20	2,43	139,00	4,19	2,83
		3	205,00	1,76	223,00	4,18	8,93
2	1	1	125,00	6,56	132,40	4,45	6,38
		2	224,80	4,75	237,60	4,82	5,22
		3	352,80	4,17	377,20	4,63	7,52
	2	1	127,40	6,30	132,60	4,53	4,20
		2	220,80	4,96	235,80	4,90	6,29
		3	352,40	4,36	379,00	4,60	7,56
	3	1	148,40	5,89	154,60	4,49	5,38
		2	217,40	4,95	226,80	4,76	3,86
		3	345,60	4,13	366,80	4,68	6,31
3	1	1	147,20	5,12	161,20	5,92	9,40
		2	255,20	4,04	283,40	6,22	10,33
		3	436,40	3,39	458,00	6,06	5,12
	2	1	172,20	5,32	182,60	6,08	6,36
		2	234,80	4,20	248,80	6,21	6,35
		3	380,80	3,49	417,20	6,28	9,74
	3	1	172,20	4,80	182,60	6,05	6,25
		2	259,80	4,17	281,20	6,32	8,05
		3	377,20	3,60	409,60	6,19	9,37
4	1	1	4024,80	5,47	4144,80	3,94	3,05
		2	7112,80	4,48	7713,20	4,26	8,43
		3	10686,00	4,04	11863,60	4,21	10,57
	2	1	4555,60	5,08	4692,40	4,16	3,08
		2	6759,20	4,59	6921,60	4,20	2,01
		3	10698,80	4,08	11320,40	4,07	6,00
	3	1	4565,20	5,49	4713,20	4,10	3,27
		2	7023,60	4,61	7076,40	4,33	0,82
		3	11288,00	4,03	11908,80	4,01	5,81
5	1	1	144,60	7,79	147,00	12,68	1,63
		2	218,40	7,13	232,20	13,25	6,00
		3	383,80	6,22	414,00	12,67	8,01
	2	1	133,00	8,01	137,80	12,52	3,59
		2	215,60	7,25	233,80	13,13	8,23
		3	382,80	6,33	408,80	12,72	7,08
	3	1	145,00	7,71	147,00	12,44	1,28
		2	204,00	7,41	217,80	13,14	6,77
		3	355,60	6,50	385,40	12,96	8,39
6	1	1	85,80	4,53	101,80	4,04	17,72
		2	143,60	3,81	153,80	4,41	6,85
		3	230,80	3,38	247,80	4,46	7,90
	2	1	91,00	4,50	105,00	3,98	14,37
		2	145,20	3,60	153,80	4,26	5,42
		3	226,40	3,29	245,60	4,39	8,71
	3	1	100,60	4,23	113,60	3,94	12,17
		2	138,00	3,84	141,20	4,14	2,12
		3	205,60	3,39	221,20	4,46	7,42

Tablo B.16: 6 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (1-6)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
7	1	1	1001,80	9,81	1029,20	10,47	3,16
		2	1478,20	6,94	1580,60	11,05	6,50
		3	2653,40	5,75	2825,60	10,95	6,33
	2	1	1097,40	8,55	1133,40	10,62	3,22
		2	1578,00	7,03	1656,00	11,12	4,77
		3	2637,80	5,60	2827,60	10,99	7,11
	3	1	1117,80	8,86	1118,00	10,63	0,02
		2	1642,00	6,94	1699,40	11,13	3,55
		3	2552,00	5,76	2713,60	10,93	6,47
8	1	1	23296,80	88,90	23735,60	12,96	1,91
		2	39471,60	20,44	44178,60	13,40	11,22
		3	51990,40	7,29	56840,00	13,32	9,51
	2	1	23574,80	104,82	24114,80	12,88	2,35
		2	38218,40	11,58	42707,00	13,28	11,20
		3	49243,20	7,04	54671,00	13,32	11,27
	3	1	24632,00	71,64	24800,40	12,94	0,65
		2	37199,20	16,13	40813,40	13,40	9,38
		3	50886,40	6,87	55735,20	13,14	9,53
9	1	1	43351,00	59,35	44878,20	21,34	3,46
		2	69574,20	372,80	73674,80	21,91	6,39
		3	112191,00	9,04	120278,60	21,61	7,19
	2	1	46416,40	30,51	48929,40	21,73	5,58
		2	66569,20	15,29	70127,80	21,89	5,28
		3	115122,80	9,04	121797,60	21,44	5,81
	3	1	49548,60	34,72	52602,60	21,78	6,09
		2	69649,40	11,19	74402,00	21,93	6,74
		3	104221,00	9,49	111517,60	21,57	7,03
10	1	1	4406,40	6,69	4590,60	7,07	4,02
		2	6379,60	6,13	6934,40	7,29	8,29
		3	10978,80	5,06	12055,60	6,96	10,05
	2	1	4580,00	6,55	4605,00	7,05	0,50
		2	6759,20	5,96	7284,40	7,32	7,57
		3	10242,80	5,13	11303,60	7,05	10,71
	3	1	4560,40	6,56	4638,40	7,01	1,85
		2	7146,00	5,87	7638,80	7,33	6,38
		3	9626,00	5,21	10442,20	7,00	8,50
11	1	1	429,80	8,77	470,00	7,40	8,98
		2	767,60	5,50	836,00	7,82	8,39
		3	1294,60	4,61	1364,20	7,58	5,54
	2	1	425,60	42,70	466,00	7,30	9,84
		2	585,20	6,72	614,60	7,77	5,32
		3	1283,60	4,47	1365,40	7,72	6,61
	3	1	504,80	9,95	528,60	7,47	4,67
		2	715,20	6,19	730,00	7,93	2,11
		3	970,80	4,84	1053,00	7,88	8,48
12	1	1	400,40	8,20	407,40	10,97	1,94
		2	665,00	5,76	711,40	11,59	6,62
		3	1092,40	5,05	1198,00	11,15	9,78
	2	1	437,60	7,42	446,20	11,30	2,19
		2	740,40	5,87	785,40	11,52	5,88
		3	1164,40	5,10	1278,80	11,09	9,83
	3	1	443,40	7,66	445,00	11,12	0,31
		2	698,60	5,77	763,60	11,60	9,09
		3	995,20	5,14	1094,20	11,26	9,99

Tablo B.17: 6 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (7-12)

Problem No	ϵ	δ	GAMS		ROHDA		Hata Oranı (%)
			Çevrim Zamanı	CPU (sn)	Çevrim Zamanı	CPU (sn)	
13	1	1	457,20	9,04	471,60	14,81	2,98
		2	789,00	7,38	846,40	15,41	6,90
		3	1418,40	6,71	1489,20	14,99	4,86
	2	1	497,40	9,02	515,60	15,05	3,80
		2	857,60	7,19	906,40	15,35	5,26
		3	1314,40	6,68	1383,40	15,16	5,62
	3	1	516,80	8,92	531,20	15,02	2,65
		2	820,20	7,49	870,40	15,43	5,91
		3	1104,40	6,93	1193,20	15,19	8,29
14	1	1	1107,80	10,18	1135,00	17,94	2,52
		2	1990,00	8,34	2134,80	18,80	6,99
		3	3135,20	7,22	3380,00	18,39	7,96
	2	1	1211,20	10,36	1221,20	17,81	0,86
		2	2097,80	8,27	2179,60	18,59	3,60
		3	3098,40	7,25	3389,60	18,37	9,59
	3	1	1350,60	10,29	1368,60	18,11	1,27
		2	1976,40	8,45	2071,60	18,77	4,79
		3	2834,60	7,29	3063,20	18,47	8,08
15	1	1	1533,60	18,60	1571,60	30,74	2,37
		2	2360,40	14,73	2535,00	31,66	7,39
		3	4314,00	13,50	4780,20	31,02	10,80
	2	1	1674,40	16,87	1734,00	30,64	3,47
		2	2537,60	14,30	2714,00	31,52	6,87
		3	3968,80	13,37	4385,00	31,12	10,49
	3	1	1747,40	16,75	1802,40	30,90	3,17
		2	2673,20	14,22	2872,00	31,68	7,32
		3	4020,80	13,28	4393,20	31,41	9,21
16	1	1	1100,00	17,92	1145,60	28,49	3,79
		2	2346,00	13,84	2610,00	30,31	11,07
		3	3859,80	13,78	4291,60	29,91	11,14
	2	1	1329,40	16,18	1366,00	29,47	2,81
		2	2260,00	13,76	2488,40	30,30	10,06
		3	3680,00	13,60	3988,20	29,62	8,47
	3	1	1463,40	16,43	1516,80	30,07	3,68
		2	2242,00	14,25	2399,40	30,31	6,92
		3	2884,40	13,48	3125,40	30,22	8,50
17	1	1	24028,00	74,13	24931,80	92,15	4,84
		2	32111,60	53,49	34493,20	95,65	7,70
		3	56745,60	53,16	60453,60	94,41	6,47
	2	1	25714,20	61,85	26662,80	94,37	4,57
		2	38033,60	52,57	40934,00	95,73	7,78
		3	49308,40	52,49	52853,80	94,44	7,18
	3	1	26178,80	60,84	27401,20	93,09	6,23
		2	34343,80	52,62	35953,60	95,99	4,92
		3	44691,20	52,08	48200,40	95,42	8,11

Tablo B.18: 6 makine $F - ratio = 0,25$ için ayrıntılı sonuçlar (13-17)

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı : ÜSTÜNOL, Sumru
Uyruğu : T.C.
Doğum tarihi ve yeri : 30.05.1988 Ankara
Medeni hali : Bekar
Telefon : 05425027950
e-mail : sustunol@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Y. Lisans	TOBB Ekonomi ve Teknoloji Üniversitesi	2012
Lisans	Eskişehir Osmangazi Üniversitesi	2010

Yabancı Dil

İngilizce (Çok iyi)