

ÖZDEŞ PARÇA ÜRETEN ESNEK İŞLEMLİ AKIŞ TİPİ
SİSTEMLERDE ÇIKTI MİKTARININ EN BÜYÜKLENMESİ

FATMA İŞLER

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AĞUSTOS 2013

ANKARA

Fen Bilimleri Enstitü onayı

Prof. Dr. Necip CAMUŞCU

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Prof. Dr. Tahir HANALİOĞLU

Anabilim Dalı Başkanı

FATMA İŞLER tarafından hazırlanan ÖZDEŞ PARÇA ÜRETEN ESNEK İŞLEMLİ AKIŞ TİPİ SİSTEMLERDE ÇIKTI MİKTARININ EN BÜYÜKLENMESİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Hakan GÜLTEKİN

Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd. Doç. Dr. Niyazi Onur BAKIR

Üye : Yrd. Doç. Dr. Hakan GÜLTEKİN

Üye : Doç. Dr. Erdem ACAR

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.

Fatma İŞLER

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 – Ağustos 2013

Fatma İŞLER

ÖZDEŞ PARÇA ÜRETEN ESNEK İŞLEMLİ AKIŞ TİPİ SİSTEMLERDE ÇIKTI MİKTARININ EN BÜYÜKLENMESİ

ÖZET

Bu çalışma kapsamında iki problem ele alınmıştır. İlk problem n adet özdeş parçanın üretildiği 3-makineli ve tek esnek işlemlili bir akış atölyesidir. Sistemdeki bütün makineler kendi sabit işleminin yanı sıra esnek işlemi de yapabilecek kapasitededir. Her parçanın esnek işlemi sadece bir makineye atanmalıdır. İkinci problemde ise yine özdeş parçaların üretildiği m -makineli, $(m - 1)$ esnek işlemin yer aldığı bir akış atölyesi ele alınmıştır. Bu problemde ise her ardışık iki makine arasında bir esnek işlem bulunmaktadır. Her iki problemdeki amaç da her parçanın esnek işleminin/işlemlerinin hangi makinede işleneceğine karar vererek üretim çıktı miktarını enbüyüklemektir. Öncelikle problemlerin karma tamsayılı matematiksel modelleri geliştirilmiştir. Ancak veri boyutu arttıkça optimal sonuca ulaşmak güçleştiği için her iki problem için de ayrı sezgisel algoritmalar geliştirilmiştir. Diğer taraftan, 3-makineli problem için kesin çözüme matematiksel modelden çok daha kısa sürede ulaşan dinamik programlama algoritması geliştirilmiştir. Geliştirilen sezgisellerin performanslarını test etmek için kapsamlı deneysel çalışmalar yapılmıştır. Matematiksel modelin sonuçları, GAMS 23.8.2 arayüzü kullanılarak CPLEX 12 çözücüsü yardımıyla elde edilmiş, sezgisel algoritma ise C++ ile kodlanmıştır. Elde edilen sonuçlar, çözüm süresi ve optimal sonuca uzaklık değerleri dikkate alınarak analiz edilmiştir.

Anahtar Kelimeler: Akış atölyesi, esnek üretim sistemleri, çizelgeleme, sezgisel yöntemler, dinamik programlama.

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. – AUGUST 2013

Fatma İŞLER

**THROUGHPUT MAXIMIZATION IN FLOWSHOPS PRODUCING
IDENTICAL PARTS HAVING FLEXIBLE OPERATIONS**

ABSTRACT

In this study, two problems are considered. The first one is a 3-machine flowshop producing n identical parts with a flexible operation. All machines in the system, performs a fixed operation on the parts and have the capability to produce an additional operation called the flexible operation. Flexible operation of each part must be assigned only one machine. The second problem is an m -machine flowshop with $(m - 1)$ flexible operations. There is a flexible operation between each of two consecutive machines. In both problems, the objective is to determine the assignment of the flexible operations of all parts in order to maximize the throughput rate. We formulated the problems as Mixed Integer Programs. However as the problem size increases, the required CPU time to solve the models increases drastically. Therefore we developed heuristic algorithms for both problems, we also developed a dynamic algorithm for the first problem which can find the optimal solution more quickly than the mathematical model. To test the performances of heuristic algorithms, comprehensive computational studies are made. We coded mathematical models using GAMS interface with CPLEX 12 Solver and we coded the heuristic algorithms in C++ programming language. The results are analyzed, with respect to required CPU times and the percent gaps with the optimal solution.

Keywords: Flowshops, flexible manufacturing systems, scheduling, heuristics, dynamic programming.

TEŐEKKÜR

Bu alıŐmayı tamamlamamda yardımlarını, özverisini ve emeğini asla esirgemeyen, hatalarımı eksiklerimi her daim anlayıŐla karŐılayarak tolere eden, her konuda bilgi ve tecrübesine başvurabildiğim tez danıŐmanım Yrd. Do Dr. Hakan GÜLTEKİN'e, tez jürisinde yer alarak tezimi okuyan ve deęerlendiren deęerli hocalarım Yrd. Do. Dr. Niyazi Onur BAKIR ve Do. Dr. Erdem ACAR'a, hayatım boyunca her zaman yanımda olduklarına emin olduęum aileme, bu süreçte desteęini asla eksik etmeyen niŐanlım Rafet SEVER'e, en güzel en içten dostluęu yaŐattıkları, iyi kötü her anımı paylaŐtıkları için baŐta Saliha ALTUNTAŐ, Aya ARIKAN, Seda SUCU ve Buęra ERSÜ olmak üzere bütün asistan arkadaşlarıma ve 110M489 numaralı proje kapsamında alıŐmalarımı destekleyen TÜBİTAK'a teŐekkürü bir bor bilirim.

İçindekiler

1 Giriş	1
2 Literatür	4
2.1 Akış Tipi Üretim Sistemlerinde Çizelgeleme	4
2.2 Üretimde Esneklik Kavramı	8
2.3 Akış Tipi Üretim Sistemlerinde Esneklik	10
2.4 Özet	14
3 3 Makineli Tek Esnek İşlemlili Akış Atölyesi	15
3.1 Problem Tanımı ve Matematiksel Model	15
3.2 Temel Sonuçlar	18
3.3 Sezgisel Çözüm Yöntemi	27
3.3.1 Algoritma adımları	28
3.3.2 Deneysel çalışma	31
3.4 Dinamik Programlama Algoritması	40
4 m-makine ($m - 1$) Esnek İşlemlili Akış Atölyesi	55

4.1	Problem Tanımı ve Matematiksel Model	55
4.2	Sezgisel Çözüm Yöntemi	58
4.3	Deneysel Çalışma	61
5	Sonuç ve Değerlendirme	67
	Kaynaklar	70
	ÖZGEÇMİŞ	74

Şekil Listesi

3.1	3 Makine Tek Esnek İşlemlili Akış Atölyesi	16
3.2	Durum 1 İçin Alt Sınır Gantt Şeması	22
3.3	$C_{max}(1)$ Gantt Şeması	23
3.4	Durum 2 İçin Alt Sınır Gantt Şeması	24
3.5	$C_{max}(2)$ İçin Gantt Şeması	25
3.6	Durum 3 İçin Alt Sınır Gantt Şeması	26
3.7	$C_{max}(3)$ İçin Gantt Şeması	26
3.8	Örnek 1 İçin Sezgisel Yöntemle Bulunan Çizelge	31
3.9	Örnek 1 İçin Optimal Çizelge	31
3.10	Dinamik Programlama Notasyonu	42
3.11	Son parçanın (2,4) Durumunda Tamamlanma Zamanı Farkı	44
3.12	3. Parçanın Atamasının Makine 1, 2 ve 3'te olduğu durum	46
3.13	Örnek 2 İçin Optimal Çizelge	48
4.1	m -makineli Akış Atölyesi	56

Tablo Listesi

3.1	İlk Parça İçin Esnek İşlem Ataması Alternatifleri	19
3.2	İkinci Parçanın Esnek İşleminin Ataması	21
3.3	Deney Faktörleri	32
3.4	Bütün Problemleri İçeren Genel Tablo	34
3.5	20 Parça İçin Özet Tablo	36
3.6	50 Parça İçin Özet Tablo	37
3.7	100 Parça İçin Özet Tablo	39
3.8	Örnek 2 İçin (a_i, b_i) 'nin Alabileceği Değerler	44
3.9	Parça 4 İçin Elde Edilen Tamamlanma Zamanı Farkları	45
3.10	Parça 3 İçin Elde Edilen Tamamlanma Zamanı Farkları	47
3.11	Örnek 2 İçin Son Tablo	47
4.1	Deney Faktörleri Dizaynı	61
4.2	5 Makine 20 İş İçin Özet Tablo	62
4.3	5 Makine 50 Parça İçin Özet Tablo	63
4.4	15 Makine 20 Parça İçin Özet Tablo	64
4.5	15 Makine 50 Parça İçin Özet Tablo	64

4.6	Bütün Problemleri İçeren Genel Tablo	65
-----	--	----

1. Giriş

Çizelgeleme birçok üretim ve hizmet sistemlerinde önemli rol oynayan bir karar verme sürecidir. Satın alma, üretim, taşıma, bilgi işleme ve haberleşme gibi çok çeşitli alanlarda kullanılır. Çizelgeleme fonksiyonu bir şirkette matematiksel veya sezgisel yöntemlerle işlerin gerçekleştirilmesinde sınırlı kaynakların tahsis edilmesine imkan sağlar. Kaynakların etkin tahsis edilmesi, şirketin amaçlarına en iyi şekilde ulaşmasını sağlar [6].

Bir çok imalat ve montaj sistemlerinde parçalar farklı makineler üzerinde işlem görürler. Tüm işlerin rotası aynı ise, yani tüm işler aynı makineleri aynı sırada takip ediyorlar ise, bu ortam akış tipi olarak adlandırılır [36].

Günümüzde teknolojinin hızlı değişimiyle, tüketiciye sunulan ürünler de çok kısa zaman içinde değişime uğramaktadır. Ürünlerin piyasada bulunabildikleri sürelerin bu kadar kısa olması ise rekabeti daha belirgin bir hale getirmektedir. Bu sebeple endüstriyel alanda değişime en kolay adapte olabilen ve üretimini etkin bir şekilde planlayıp çizelgeleme problemlerine kısa sürede çözüm üretebilen firma bir adım öne geçme şansına sahiptir. Ancak sektörlerin bu değişimin hızına yetişebilmek için sürekli üretim sistemlerini yenilemek gibi bir lüksü yoktur. Bu sebeple mevcut sistemin daha etkin kullanılabilmesi ve farklı durumlara uygulanabilirliğinin olması gerekmektedir. Tam bu noktada esnek üretim sistemleri karşımıza çıkmaktadır. Esnek Üretim sistemleri belli bir ürün ya da modelle sabit kalmayıp farklı üretim alternatifleri sunabilecek kabiliyete sahip sistemlerdir.

Bu çalışmada 3 makineli ve m -makineli olmak üzere iki problem ele alınmıştır:

- 3 makineli akış tipi atölye probleminde, n adet özdeş işin üretildiği

varsayılmaktadır. Parçalar makinelerden sırasıyla geçerken her makinede bir işleme tabi tutulmaktadır. Her makinenin kendine özgü sabit işlemlerinin yanı sıra parçaların ürün haline gelebilmesi için bir de esnek işlem den geçmesi gerekmektedir. Sistemdeki her üç makine de bu esnek işlemi yapabilme kabiliyetine sahiptir. Problemden her parça için bu esnek işlemin hangi makinede yapılacağı kararının verilmesi gerekmektedir. Problemden amaç son parçanın makine 3'teki tamamlanma zamanını en küçüklemeğektir.

Problemden makineler arası ara stok alanı sınırsız kabul edilmektedir. Bu nedenle çözüm yöntemlerinde bu bir kısıt olarak alınmamıştır. Sınırsız ara stok kapasitesi, n parçanın üretildiği bir sistemde $n - 1$ adet ara stok alanı bulunması durumunda geçerlidir. Çünkü bu durumda makinelerin ara stok alanında yer kalmayıp parçanın makine üzerinde kalarak o makinenin çalışmasını engellemesi (bloke olma) durumu hiçbir zaman gerçekleşmeyeceğektir.

Bu problemin çözümü için öncelikle karma tamsayılı matematiksel model kurulmuştur. Parça sayısı ya da makine sayısı gibi problem verilerindeki artışla matematiksel modelin sonuç verme süresinin oldukça arttığı gözlemlenmiştir. Gerçek hayatta bir üretim sisteminde, her gün en az bir kere karar verilmesi gereken bir problem olduğu için matematiksel modelin sonuç verdiği sürelerin kabul edilebilir zamanlar olmadığı görülmüştür. Bu nedenle problemin çözümü için sezgisel yöntem geliştirilmiş ve geliştirilen algoritmanın test edilmesi için deneysel çalışma yapılmıştır. Sezgisel yöntemle geliştirilen algoritma ile oldukça iyi sonuçlar elde edilmiş olmasına rağmen sadece 3 makinenin bulunduğu bir sistem için optimal sonuçların dinamik programlama ile bulunabileceği öngörülerek bir dinamik programlama algoritması geliştirilmiştir. Ancak problem verilerinin dinamik programlama algoritmasının çözüm süresinde de önemli bir kriter olarak karşımıza çıktığı görülmüştür. Bu nedenle dinamik programlama algoritmasının çalışma süresini kısaltacak çeşitli teoremler ispatlanmıştır.

- Ele alınan bir diğer problem ise m -makineli sistemdir. Sınırsız ara stok alanı, n parçanın üretilmesi ve her makinenin kendi sabit işleminin olduğu varsayımı bu problem için de geçerlidir. Ancak bu sistemde ilk makineden başlamak üzere her ardışık iki makine arasında bir esnek işlem bulunmaktadır. Her parçanın ürün haline gelebilmesi $m - 1$ adet esnek işleme tabi

tutulması gerekmektedir. Bu problemdeki amaç da benzer şekilde esnek işlemlerin hangi makinelerde yapılacağına karar vermektir. Her parçanın her esnek işlemi için karar verilerek son parçanın m makinesindeki tamamlanma zamanı enküçüklenmeye çalışılmaktadır.

Bu problemin çözümü için de 3 makineli probleme benzer şekilde karma tamsayılı matematiksel model geliştirilmiştir. Beklendiği üzere çözüm süresi küçük problem verilerinde dahi çok büyük olduğu durumlarla karşılaşıldığı için bu problem için de bir sezgisel yöntem geliştirilmiştir. Deneysel bir çalışma yapılarak geliştirilen yöntem test edilmiştir.

Tez çalışması 5 bölüm olarak incelenmiştir. Bir sonraki bölümde literatür araştırmasına yer verilerek, akış tipi üretim sistemlerinde çizelgeleme, üretimde esneklik kavramı ve akış tipi üretim sistemlerinde esneklik konularında yapılmış çalışmalar taranmıştır. Bölüm 3 ve Bölüm 4'te ise sırasıyla 3 makineli ve m -makineli akış atölyesi problemleri ele alınmıştır. Her iki bölüm de kendi içinde problem tanımı, çözüm yöntemleri ve deneysel çalışma olmak üzere alt bölümlere ayrılmıştır. Son bölümde ise, çalışmayla ilgili sonuçlar ve yorumlara aynı zamanda gelecek çalışmalarda ne gibi problemlerin ele alınabileceğine yer verilerek çalışma tamamlanmıştır.

2. Literatür

Bugüne kadar çizelgeleme konusunda yapılmış çok sayıda çalışma vardır. Makine çizelgeleme problemlerini; tek makine çizelgeleme problemleri, paralel makine çizelgeleme problemleri, akış tipi çizelgeleme problemleri, atölye tipi çizelgeleme problemleri gibi başlıklara ayırmak mümkündür. Bu çalışmanın da içeriğini oluşturan akış tipi çizelgeleme problemleri, araştırmacıların yoğun olarak çalıştığı optimizasyon problemlerinden biridir. Ancak gün geçtikçe teknolojinin gelişimiyle klasik akış tipi çizelgeleme problemleri de birçok yeni alt başlığa ayrılmaktadır. Bunlardan en önemlisi çok değişken müşteri taleplerini kısa sürede ve düşük maliyetle karşılamayı sağlayan esnek üretim sistemleridir.

Akış atölyesi çizelgeleme literatürü 1950'li yıllara dayanmaktadır. Üretimde kullanılan sistemlerin değişimine paralel olarak literatürdeki çalışmalar da zamanla değişmiştir. Bu çalışma klasik akış tipi üretim sistemlerinden farklı olarak esnek makinelerin kullanıldığı bir üretim sistemini ele almaktadır. Yapılan çalışmanın literatürdeki yerini daha iyi görebilmek amacıyla bugüne kadar yapılmış çalışmalar akış tipi üretim sistemleri, üretimde esneklik kavramı, akış tipi üretim sistemlerinde esneklik ve montaj hattı dengeleme başlıkları altında incelenecektir.

2.1 Akış Tipi Üretim Sistemlerinde Çizelgeleme

Akış tipi üretim sistemlerinin temelini her parçanın bütün makinelerde aynı sırayı izleyerek işlenmesi oluşturur. Parçaların özdeş olması ya da olmaması, işlem sürelerinin değişkenlik göstermesi ya da sabit kalması gibi çokça kriterin dikkate alınmasıyla bu sistemleri ele alan problemler çeşitlendirilebilmektedir.

Bu konuyla ilgili daha geniş bilgiye Pinedo [36]'dan ulařılabilir. Genel olarak akıř tipi üretim sistemleri üzerine çok sayıda alıřma yapılmıřtır. Bu konunun temelini oluřturan alıřma Johnson [26]'ın 1954 yılında literatürde yerini almıř olan alıřmasıdır. İki makinenin yer aldıđı farklı tip paraların üretildiđi sistemde Johnson optimal çözüme ulařan bir algoritma geliřtirmiřtir. Bu algoritma iřlerin hangi sırayla iřleneceđini bulmaktadır. Sistemin sonsuz ara stok kapasitesi olduđu varsayılmaktadır. Her iřin makinelerdeki iřlem zamanları bellidir. Bařlangı anında her iřin hazır olduđu da varsayımlar içindedir. Algoritmanın iřleyiři řu şekildedir:

- Her paranın iki makinedeki iřlem zamanlarını listele.
- Bütün iřlem zamanları arasında en küçük olanı se.
- Belirlenen bu iřlem süresi; ilk makineye aitse bu iřlem süresine sahip iři ilk sıraya, ikinci makineye aitse bu iři son sıraya yaz.
- Belirlenen iři listeden çıkar.
- Bu iřlemi bütün paralar sıralanana kadar tekrarla.

Algoritma $O(n \log(n))$ alıřma süresine sahiptir. Ü makine farklı tip para probleminin ise NP-Zor olduđu Garey ve Johnson [16] tarafından ispat edilmiřtir. Ancak 3 makineli sistemin özel bir durumu olan ortadaki makinenin en büyük iřlem süresinin diđer iki makinenin en küçük iřlem süresinden küçük olduđu durumda yukarıdaki algoritma modifiye edilerek optimal sonucu vermesi sađlanmaktadır [26].

Akıř tipi üretim sistemlerinin geniř aplı taraması Hejazi ve Saghafian [24] tarafından yapılmıřtır. Makalede yayılma zamanının enküçüklenmesine yönelik alıřmalarının yanı sıra geliřtirilen temel algoritmalara ve metasezgisellere de yer vermiřlerdir. Ayrıca ayar zamanlarının çeřitlerine göre problemlerin sınıflandırıldıđı akıř tipi sistemler için bir bařka tarama makalesi de Cheng vd. [11] tarafından ele alınmıřtır. Bu alıřmada makaleler bađımlı ve bađımsız ayar zamanlarına göre, aile ayar zamanı ve para ayar zamanı olmalarına göre farklı alt bařlıklar altında incelenmiřtir. Ayar zamanlarının ele alındıđı bir diđer alıřma ise Allahverdi vd. [2] tarafından ele alınmıřtır.

m makinenin yer aldığı NP-zor olduğu bilinen sistemler için geliştirilen çok sayıda algoritma vardır. Literatürde önemli yere sahip başlıca 4 algoritma geliştirildikleri tarihlere göre şöyle sıralanabilir.

- **Palmer Algoritması(1965):** Küçük işlem süresine sahip işin önce işlenmesi prensibine dayanan bu algoritma da her iş için makine sayısına bağlı bir eğim indeksi hesaplanır. Bu indeksler büyükten küçüğe sıralanarak bir sıralama elde edilir [35].
- **CDS Algoritması (1970):** m makinenin olduğu bir sistemi $m - 1$ tane iki makineli sisteme dönüştürerek elde edilen problemlerin Johnson algoritmasıyla çözüldükten sonra en iyisinin seçildiği bir algoritmadır [10].
- **Gupta Algoritması (1971):** Bu algoritma da Palmer Algoritması gibi çalışmaktadır. Eğim indeksini farklı bir formülle hesaplayarak daha iyi sonuca ulaşmıştır [22].
- **NEH Algoritması(1983):** Nawaz vd. [32] tarafından geliştirilmiştir. Daha sonradan geliştirilen birçok iyileştirme algoritmasına temel oluşturmaktadır. NEH algoritmasının adımları şöyledir:
 - **Adım 1:** Her bir işin bütün makinelerdeki işlem zamanlarının toplamını bul.
 - **Adım 2:** En büyük toplam işlem zamanına sahip ilk iki işi sıralarını değiştirerek çizelgele ve küçük tamamlanma zamanını veren sıralamada sabitle.
 - **Adım 3:** Yeni eklenecek her iş için çizelgede yerleşebileceği her sırayı dene ve en küçük tamamlanma zamanını verdiği değerdeki yere yerleştir.
 - **Adım 4:** Bu işlemi n . işe kadar tekrarla.

Bu algoritma permütasyon akış tipi çizelgeleme problemleri için en etkili çözüm yöntemidir. Permütasyon akış tipi çizelgeleme problemi, tüm makinelerde bir işin işlem sırasının aynı olduğu, m makine ($j = 1, 2, \dots, m$) üzerinde belli işlem sürelerine sahip n işin ($i = 1, 2, \dots, n$) çizelgenmesinden oluşan problemlerdir. Algoritmanın performansı test edilirken yapılan deneysel

çalışmalar neticesinde $n = 5, \dots, 500$ ve $m = 5, \dots, 25$ olduğu aralıkta optimale yakın sonuçlar verdiği gözlemlenmiştir [32].

Akış tipi çizelgeleme problemlerinde, Johnson [26]'dan sonra yapılan çalışmaların incelendiği bir diğer tarama makalesi ise Gupta ve Stafford [21] tarafından ele alınmıştır. Ele alınan problemlerin zamanla nasıl değişim gösterdiğini incelemişlerdir. Bu süreçte, ele alınan problemlere, gerçek hayatta karşılaşılan zorlukların entegre edilmesiyle problemler farklı karakteristik özellikler kazanmaya başlamışlardır. Ara stok kapasitesine ve işlem özelliklerine göre akış tipi sistemleri *sınırsız ara stok*, *bloke olma* ve *beklemesiz sistemler* başlıkları altında incelemek mümkündür. Bu kavramları sırasıyla ele aldığımızda ilki olan *sınırsız ara stok* kavramı bu tez çalışmasında ele alınmıştır. Üretim esnasında makineler arasındaki stok alanının üretilen toplam parça sayısının bir eksiği kadar veya daha fazla olması durumudur. İlerleyen bölümlerde bu kavramdan sıkça bahsedilecektir. Bir diğer kavram *bloke olma* ise, bir parçanın işlemi bir makinede tamamlandıktan sonra bir sonraki makineye geçeceğinde eğer ikinci makinenin üzerinde hala işlenmekte olan bir parça varsa ara stok bulunmadığı ya da ara stokta yer olmadığı durumlarda söz konusu parça ikinci makinedeki işlem tamamlanana kadar ilk makinede beklemelidir. Bu durumda ilk makine işlem yapamadığı için bloke olmaktadır. *Beklemesiz sistemler* ise akış tipi makine ortamında, işlerin birbirini takip eden iki makine arasında beklemeyeceğini ifade eder. Hall ve Sriskandarajah [23] tarafından *bloke olma* kavramıyla birlikte *beklemesiz ortam* kavramının da yer aldığı çalışmaları özetleyen bir tarama makalesi daha bulunmaktadır.

Gangadharan ve Rajendran [15], akış tipi sistemde, yayılma zamanı ve toplam akış zamanını enküçükleme için tavlama benzetimi algoritması kullanmışlardır. Problemi 25 makine 40 işe kadar çözüp, elde ettikleri sonuçları Ho ve Chang [25] ile Ogbu ve Smith [34]'in sonuçları ile karşılaştırarak geliştirdikleri sezgisel ile daha etkin sonuçlar bulduklarını göstermişlerdir. Yeh [41] tarafından ele alınan çalışmada ise iki makineli akış tipi çizelgeleme problemi için, toplam akış zamanı ve yayılma zamanı amaç fonksiyonları ele alınmıştır. Geliştirilen dal-sınır algoritması ile 200 işe kadar çözümler elde edilmiş ve daha önce geliştirilen bazı algoritmalarla karşılaştırılmıştır.

Literatürde çok fazla çalışmanın yer aldığı akış tipi üretim sistemlerinin özel bir kolu ise montaj hatlarıdır. Montaj hattı problemleri de akış tipi problemlerinin bir koludur [7]. Montaj hattı, bir ürünün hattın başından sonuna kadar her aşamada farklı bir katkı sağlanarak tamamlanmasını ifade etmektedir. Montaj hattının ortaya çıkışı 1900'lü yıllarda Henry Ford ile olmuştur. İlk montaj hattı örneği T tipi montaj hattıdır [8]. Günümüze kadar montaj hatlarının kullanımı oldukça yaygınlaşmış ve doğan ihtiyaçlara cevap verebilmek için çeşitlenmiştir. Başlangıçta tek model üreten ve sabit tempolu işlem yapan bu hatlar zamanla özelleşmiş paralel istasyonlu hatlara, ara stok barındıran hızı değişebilen U tipi hatlara dönüşmüştür. Ancak bu dönüşüm kolaylıklarıyla birlikte iş yükünün hat boyu dağılması problemi olan montaj hattı dengeleme kavramını da doğurmuştur [7].

Montaj hattı dengeleme problemlerinde hattaki boş zaman miktarı enküçüklenmeye çalışılır. Hat dengelemede çevrim süresi ve istasyon sayısı kavramları ele alınır. İki farklı amaç olabilir. İlk üretim hızının enbüyüklenmesidir ki bunun için ele alınacak olan çevrim süresinin küçültülmeye çalışılmasıdır. Bunun için istasyon sayısının sabit olduğu varsayılır. Diğer amaç ise istasyon maliyetlerini en aza indirmektir. Bunun için ise çevrim zamanı verilmişken istasyon sayısı en aza indirilmeye çalışılır. Hattın tipine ve ürüne göre bu performans ölçütlerinden birinden yararlanılarak hat dengeleme yapılabilir.

2.2 Üretimde Esneklik Kavramı

Esneklik kavramının üretimdeki yerini daha iyi anlayabilmek için bu bölümde esneklik tanımına, çeşitlerine, esnekliğin katkılarına yer verilmektedir.

Son yıllarda esneklik kavramını ele alan oldukça fazla çalışma yapılmıştır. Çalışmaların çeşitliliği esnekliğin tek bir tanımla sınırlandırılmayacağını da göstermiştir. Genel olarak esneklik tanımı farklı durumlara adapte olabilme yeteneği olarak tanımlanabilmektedir. Üretimde esneklik ise, üretim kaynaklarının kaliteli ve farklı ürünler üretmek için şekillendirilmesidir [39].

Literatürde yer alan esneklik tanımları incelendiğinde aslında bir çoğunun

aynı kavramdan bahsettiğini söylemek mümkündür. Çalışmaların büyük kısmı esneklik tanımına ve çeşitlerine ağırlık vermiştir. Bu konuda ele alınan en önemli çalışmalardan birisi Sethi ve Sethi [39] tarafından yayınlanan bir makaledir. Esneklik konusunda önemli yeri olan bu makalede esneklik 11 alt başlığa ayrılmıştır. Bunlar, makine, malzeme taşıma, ürün, program, market, süreç, operasyon, rota, hacim, üretim ve büyüme esnekliğidir. Bu tanımları birbirinden net olarak ayırmak mümkün değildir. İç içe geçmiş ve birbirini tamamlayan niteliktedirler. Bu çalışmada ele alınan problemdeki esneklik hem makine esnekliği hem de operasyon esnekliği tanımına uygundur. Bu iki tanım daha detaylı incelenecektir.

Makine Esnekliği: Bir makinenin bir operasyondan diğerine çok düşük maliyetle ve kısa sürede geçebilmesidir. Bu şekilde farklı tipteki işlemleri yapabilme kabiliyeti olarak açıklanabilir. Makine esnekliğinin karşılaştığı sistemler nümerik kontrol (NC) ve bilgisayarlı nümerik kontrol (CNC) olarak örneklendirilebilmektedir.

Operasyon Esnekliği: Parçanın bazı operasyonlarının alternatif farklı yollarla da yapılarak parçanın üretilebileceği anlamına gelmektedir. Operasyon esnekliği aynı zamanda diğer esneklik kavramlarına da katkı sağlamaktadır. Bu esneklik çeşidi ise daha çok bilgisayarlı üretim sistemlerinde karşılaşılmaktadır.

Ayrıca operasyon esnekliğinin parçaların çizelgelenmesini kolaylaştırdığı bilinmektedir [9].

Günümüze kadar esnekliğin sınıflandırılması, hangi tip üretim sistemlerinin hangi esnek üretim sistemi sınıfı içinde değerlendirileceği, esnekliğin sisteme ne tür katkı sağlayacağını gösteren bir çok çalışma yapılmıştır. Browne vd. [9] esnek üretim sistemlerini esnek işlem hücresi, esnek işlem sistemi, esnek transfer hattı ve çoklu transfer hattı olmak üzere 4 başlık altında toplamışlardır. Bununla birlikte esnekliğin katkılarını inceleyen çalışmalardan birisi de Askin ve Chen [4]'ün iş sıralamasının bilindiği 2 makineli bir sistemi ele aldıkları çalışmadır. İşlerin farklı istasyonlarda işlenebilmesi çapraz eğitilmiş işçilerle sağlanmaktadır. Bu esnekliğin hatta denge sağlanması ve çıktı miktarını artırması beklenmektedir. Sistemde ara

stok bulunabilmektedir. Çalışmada bir benzetim modeli oluşturulmuş ve sonuç olarak ara stok miktarı azaltılarak çıktı miktarının artırılabilirdiği gözlemlenmiştir.

Mc Clain vd. [30] esnek işlemlerin stok alanı bulunmaması durumunda bile verimliliği arttırdığını göstermiştir. İki makineli akış tipi sistemi ele alan bir başka çalışma Sayın ve Karabatı [38] tarafından yapılmıştır. İki makineli sistemde yayılma zamanı ve toplam tamamlanma zamanı problemini çözmüşlerdir. Problemi 28 işe kadar çözen bir dal-sınır algoritmasını, Klein ve Hannan [29]'ın geliştirdiği çok amaçlı tamsayı doğrusal programlama modelinin özelliklerinden faydalanaarak geliştirmişlerdir. Kimemia ve Gershwin [28] maksimum akışı sağlayabilmek için esnek üretim sistemlerinde parça rotalarının eniyilenmesi problemi üzerinde durmuşlardır.

Esnek olmayan üretim sistemlerinde karşılaşılan problemlerle birlikte sistemlere ne tür esneklikler kazandırılabilirdiği de yine araştırma konusu olmuştur. Bu konuyla ilgili olarak Jordan ve Graves [27] bir çalışma yapmıştır. Esnekliğin sistemlerde uygulanamama nedenlerini ise şöyle özetlemişlerdir. İlk olarak, esnekliğin uygulanması için katlanılacak maliyet ölçülebilirken esnekliğin sağlayacağı katkı ölçülememekte ve bu da işletme sahiplerine gereksiz bir maliyete katlanmak gibi görünmektedir. İkinci sebep ise benzer şekilde esnekliğin tüm sisteme uygulanması gerektiğiyle ilgili yanlış bir algıdır. Bu da yine yüksek bir maliyet getireceği gerekçesiyle esnek sistemlerin uygulanmamasına sebep olan bir diğer etmen olarak bu çalışmada belirtilmiştir. Bu problemin farkında olarak yapılan bir diğer çalışma ise Nomden ve van der Zee [33]'ye aittir. Bu çalışmaların ortak noktası firmaların düşündüğü gibi büyük yatırım maliyetleri gerektirmeden sistemlerin esnekleştirilebileceği ve elde edilecek çıktının büyük katkı sağlayacağıdır.

2.3 Akış Tipi Üretim Sistemlerinde Esneklik

Akış tipi üretim sistemlerinde esneklik konusu literatürde yeni yeni ele alınmaya başlamış bir konudur. Esnekliğin bu sistemlere adapte olması operasyon ya da makine esnekliği söz konusu olduğunda, bir işlemin birden fazla makine tarafından yapılabiliyor olması dolayısıyla aynı işin farklı yollarla üretilmesi anlamına

gelmektedir. Bu kavramlar sistemde üretimin daha kolay yapılabilmesini ve etkin bir çizelge ile daha kısa sürede daha fazla çıktı elde edilmesini sağlayacaktır. Esnekliğin yer aldığı akış tipi sistemlerle ilgili çalışmalar son yıllarda hız kazanmıştır. Bu bölümde özellikle bu çalışmalardan bahsedilecektir.

Vairaktarakis ve Lee [40]'nin çalışmasında ele alınan sistem iki makinenin yer aldığı sistemdir. Sistemde iki operasyon yer almaktadır. İlk operasyon ilk makine ikinci operasyon ikinci makinede işlenebilmektedir. Ayrıca her iki operasyon da her iki makine tarafından yapılabilir. Sistemi sıradan bir akış tipi sistemden ayıran faktör, her iki operasyonun da aynı makinede yapılması durumunda parçanın diğer makineye hiç uğramıyor oluşudur. Problemden amaç yayılma zamanını en küçükmektir. Bunun için işlerin hangi sırayla işleneceğine ve işlemlerin makinelerde nasıl yapılacağına karar verilmesi gerekmektedir. Optimal çözüme ulaşmak amacıyla dinamik programlama yöntemi geliştirilmiştir.

Kaynak esnekliğinin ele alındığı Daniels ve Mazzola [13]'nin çalışmasında n adet iş m makinenin bulunduğu bir sistem incelenmiştir. Çalışmada kaynak esnekliğinin çapraz eğitilmiş işçilerle ya da esnek makinelerle sağlanabileceği belirtilmiştir. Çapraz eğitilmiş işçilerin birbiri yerine işlem yapabilmeleri bu çalışmada kaynak esnekliği kavramına karşılık gelmektedir. İşlerin sıralamasını belirlemek ve kaynak atamasını yapmak problemdeki amaçtır. Çözüme ulaşabilmek için sezgisel ve optimal çözüm algoritmaları geliştirilmiştir. Makine sayısı, işlem sayısı ve parça sayısı değiştirilerek deneysel bir çalışma yapılmıştır. Kaynak esnekliğinin problem çözüm performansına etki ettiği ve sezgisel yöntemle büyük boyutlu problemler için etkin çözümler elde edilebileceği sonucuna varılmıştır.

Daniels vd. [14]'nin ele aldığı bir diğer çalışma yine n iş ve m makinenin bulunduğu bir sistemde işçi esnekliğine odaklanılarak incelenmiştir. Akış tipi sistemde çapraz eğitilmiş işçilerin katkısını ölçmeyi hedeflemişlerdir. Dal-sınır algoritması geliştirilerek optimal sonuca ulaşmışlar ancak problem boyutu büyüdükçe çözüm süresi arttığı için sezgisel bir yöntem geliştirmişlerdir. Deneysel çalışma sonucunda çapraz eğitime yapılacak daha küçük maliyetlerle sistemde büyük katkıların sağlanabildiği görülmüştür.

Esnek akış tipi bir sistemin bulunduğu bir problem Babayan ve He [5] tarafından ele alınmıştır. Sistemde üretim 3 basamakta tamamlanmaktadır. Her basamakta

işlem özdeş makinelerden biri tarafından yapılabilmektedir. Amaç tamamlanma zamanını enküçükmektir. Ajan-temelli bir yaklaşım ile çözümler elde edilmeye çalışılmıştır.

Anuar ve Bukchin [3], akış tipi üretim sistemine esneklik eklendiğinde çevrim zamanının nasıl etkileneceğini incelemişlerdir. Hatlar arasında ortak yapılabilen esnek işlemlerin bulunması hatların dengelenebilmesini sağlamaktadır. Esnek işlemler bir istasyonun son işlemi veya bir sonraki istasyonun ilk işlemi olmaktadır. Problemden esnek işlem sayısına ve atamalarına karar verilmesi gerekmektedir. Kısa sürede çözüm veren algoritmalar geliştirilmiştir. Makine sayısı ve ara stok alanının farklı sayısal değerleri için deneysel bir çalışma yapılmıştır. Esnek işlemler için önerilen yaklaşım klasik akış tipi sistemlere göre daha kısa çevrim zamanları elde edilmiştir.

Operasyon esnekliğini Aktürk vd. [1] özdeş parçaların üretildiği iki makine robotik hücre çizelgeleme probleminde ele almıştır. Sistemdeki makineler tüm işlemleri yapabilme kabiliyetine sahiptir. Çalışmanın amacı operasyon atamasına ve optimal robot döngüsüne karar vererek minimum çevrim zamanını elde etmektir. Yine benzer bir çalışma Gültekin vd. [17] tarafından ele alınmıştır. Bu çalışmada kesici uç haznesi kısıtı göz önünde bulundurulmuştur. Bu kısıtla birlikte bazı işlemler sadece ilk, bazı işlemler ise sadece ikinci makinede işlem görebilmektedir. Kalan diğer işlemler, her iki makinede işlem görebilmektedir. Problemin çözümü her iki makinede de yapılabilen işlemlerin atanmasıyla ilgilidir. Çalışmada her döngü için optimal parametre aralıkları belirlenmiş ve duyarlılık analizi yapılmıştır.

Gültekin vd. [18] 2 ve 3 makineli akış atölyesinde özdeş parçaların üretildiği bir problemi ele almışlardır. Malzeme taşımanın robotlar tarafından gerçekleştirildiği bu sistemde, çevrim zamanı ve toplam üretim maliyeti performans ölçütleri dikkate alınmıştır. Robot hareket sırası ve işlem zamanlarına karar vermişlerdir. Gültekin [19] tarafından ele alınan bir başka çalışma esnek işlemlerli iki makineli akış tipi üretim sisteminde çıktı optimizasyonudur. Sistemde üretilen parçalar özdeştir. Ürün haline gelebilmesi için parçaların üç işlemden geçmesi gerekmektedir. İşlemlerden biri ilk makinede, diğeri ikinci makinede işlenebilmektedir. Esnek işlem ise her iki makinede de yapılabilmektedir. Sistemde makineler özdeş olmadığından işlem süreleri makineye göre farklılık gösterebilmektedir. Ara

stok alanının hem sıfır ve hem sınırsız olduğu durumlar için çözüm yöntemleri sunulmuştur. Çalışmada ayrıca esnekliğin yararları gösterilmiştir. Çok küçük esneklik seviyesinde dahi, daha düşük tamamlanma zamanını elde edildiği sonucuna varılmıştır.

Tez çalışmasıyla yakından ilgili olan bir çalışma Gupta vd. [20] tarafından ele alınmıştır. 2 makinenin bulunduğu sistemde n adet farklı tip parça 3 farklı işlemden geçerek üretilmektedir. İlk işlem ilk makinede, üçüncü işlem ikinci makinede işlenmekte, esnek olan ikinci işlem ise iki makine tarafından da yapılabilir. Tamamlanma zamanının en küçüklenmeye çalışıldığı problemde esnek işlemin hangi makinede yapılacağına karar vermeye çalışılmaktadır. Bununla birlikte problemin NP-zor olduğu da ispat edilmiştir. Polinom zamanlı iki sezgisel algoritma geliştirilmiş ve en kötü durum performansları belirlenmiştir. İlk algoritma esnek işlemlerin atamalarını rastgele yapmaktadır. İkinci algoritma ise 4 farklı çizelge oluşturmakta ve bunlardan en iyisini seçmektedir. Ayrıca optimal çözüme belli bir uzaklıkta çözüm bulma garantisi olan bir yaklaşım algoritması geliştirmişlerdir.

Problemde yer alan esneklik tanımı tez çalışmasıyla uyumludur. Esnek makine tanımı ve operasyonun atama kararı tamamen benzerlik göstermektedir. Tez çalışmasının problemde farkı ele alınan makine sayısının 3 olması ve özdeş parçaların üretiminin yapıyor olmasıdır. Ancak bu problemde 3 makine olması problemi biraz daha zor kılacakken özdeş parça üretiliyor olması ise kolaylaştıracaktır. Esnek işlemler, 2 makine farklı tip parça üreten sistemlerin zayıf NP-zor olduğu Gupta vd. [20] tarafından ispat edilmiştir. 3 makine farklı tip parça üreten akış tipi sistem için de NP-zor olduğu da Garey ve Johnson vd. [16] tarafından gösterilmiştir. Bu çalışmada ele alınan problem ise 3 makineli bir sistemin ele alınması ancak özdeş parça üretiliyor olması sebebiyle NP-zor sınıfına girip girmediği bilinmemektedir.

Tez çalışmasına temel oluşturan bir diğer çalışma ise Crama ve Gultekin [12] tarafından ele alınmıştır. Özdeş parçaların üretildiği 2 makineli bir akış tipi sistem incelenmiştir. Sabit işlem olarak adlandırılan ilk işlem sadece ilk makinede, ikinci işlem ise sadece ikinci makinede yapılabilir. Esnek işlem her iki makine tarafından da işlenebilir. Problemdeki amaç çıktı miktarını enbüyükleyecek şekilde esnek işlem ataması kararlarını vermektir. Problemin

farklı karakteristik özellikleri için optimal çözümün farklı özellikleri belirlenmiştir. Parça sayısının sonlu veya sonsuz olması, ara stok miktarının sıfır, sonlu veya sonsuz olması durumları ayrı ayrı ele alınmıştır. Her alt problem için optimal çözümü veren yöntemler polinom zamanlı yöntemler geliştirilmiştir. İki parça ileriye bakarak esnek işlem atamasına karar veren bir algoritma geliştirilmiş ve bu problem için optimal çözüme ulaştığı ispatlanmıştır.

2.4 Özet

Bu bölümde, akış tipi üretim sistemlerine ve esneklik konusunun üretimde ele alınış şekline, esneklik tiplerine değinilmiş ve tez çalışmasında ele alınan problemle benzerlik gösteren ve problemi anlamak açısından katkısı olduğu düşünülen literatürdeki diğer çalışmalara yer verilmiştir.

Gerçek hayat problemlerinin de ele alınması ve esnekliğin gitgide önem kazanmasıyla bu konular üzerine yapılan çalışmalar da hızla artmıştır. Problem tipleri gün geçtikçe farklılaşarak çalışmalara daha spesifik problemler konu olmaya başlamıştır. Bu bölümde bu çalışmaların bir kısmına yer verilmiş, ele alınan problemlerden ve çözüm yöntemlerinden bahsedilmiştir. Tez çalışmasında ele alınan problemin konusu Crama ve Gültekin [12] ile Gupta vd. [20] tarafından ele alınan problemlerle oldukça benzerdir. 2 makineli bir sistemde sırasıyla farklı tip parçaların ve aynı tip parçaların üretimini incelemişlerdir. Bu çalışmada ise 3 makineli bir sistem ve aynı tip parçalar ele alınmıştır. Önceki çalışmalar dikkate alınarak çözüm yöntemleri geliştirilmeye çalışılmıştır.

Bu çalışmanın klasik bir akış tipi sistemi olmaması ve günümüzde önem kazanan esneklik konusunu barındırması çalışmayı daha önemli kılmaktadır. İhtiyaçların ve zevklerin çok farklılık gösterebildiği ürünler için esnek sistemlerin kullanılması şart hale gelmiştir. Bu nedenle bu çalışmanın literatürde yer alan bir boşluğu dolduracağı ve gerçek hayat problemlerine ışık tutacağı düşünülmektedir.

3. 3 Makineli Tek Esnek İşlemlili Akış Atölyesi

Bu bölümde, 3 makineli tek esnek işlemlili akış atölyesi çizelgeleme problemine ve geliştirilen çözüm yöntemlerine yer verilmektedir. Ayrıntılı olarak problemin özelliklerine değinilerek kurulan karma tamsayılı model anlatılmakta, ardından çözüm yöntemleri geliştirilirken faydalanılan önteorem ve teoremler problem sonuçları alt başlığında sunulmaktadır. Çözüm yöntemi olarak öncelikle bir sezgisel algoritma geliştirilmiş, sonrasında ise dinamik programlama tabanlı bir kesin çözüm yöntemi önerilmiştir.

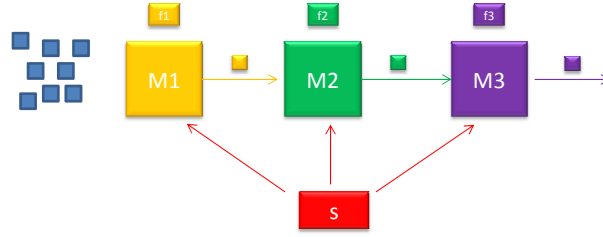
3.1 Problem Tanımı ve Matematiksel Model

Bu bölümde 3 makineli bir akış atölyesi ele alınmıştır. Problemin klasik bir akış tipi sisteminden farkı sistemde esnek işlem bulunmasıdır. Özdeş n adet parçanın üretildiği sistemde her parça üzerinde dört farklı işlem yapılarak ürün haline gelmektedir. Bu dört işlemde üç tanesi sabit işlemdir ve sırasıyla her biri farklı bir makinede yapılmaktadır. Bunlara ek olarak bir de tüm makineler tarafından yapılabilen bir esnek işlem vardır ve bu işlemin hangi makinede yapılacağına karar verilmeye çalışılmaktadır. Yayılma zamanı, n . parçanın 3. makinedeki tamamlanma zamanıdır. Akış tipi sistemin gereği olarak parçalar sırasıyla makine 1, makine 2 ve makine 3'te aynı sırayla işlenmektedir.

Problemde iş kesmeye izin verilmemektedir. Yani bir iş bir makine üzerindeki işlemi tamamlanmadan o makineden ayrılamaz. Ayrıca her esnek işlem atandığı makinede sabit işlemde hemen sonra yapılmaktadır. Parametreler deterministik

olup önceden bellidir. Bir diğer varsayım ise makinelerin özdeş olduğudur. Bu varsayımla esnek işlem hangi makineye atanırsa atansın aynı işlem süresine sahip olacaktır.

Esnek işlem her parça için sadece bir makinede yapılmalıdır. Esnek işlemin hangi makinede yapılacağı kararı her parça için birbirinden bağımsız olarak verilmektedir. Şekil 3.1’de sistemin işleyişi gösterilmiştir.



Şekil 3.1: 3 Makine Tek Esnek İşlemlili Akış Atölyesi

Sistemde makineler arasında belli bir stok alanı bulunabilir. Ara stok alanının üretilecek parça sayısının bir eksiğinden $(n - 1)$ daha fazla olması durumunda sınırsız ara stok alanı varmış gibi düşünülebilir. Çünkü bir parçanın makine üzerinde işlemde olduğu düşünülduğünde geriye kalan $n - 1$ parça için bekleyebilecekleri bir ara stok alanı mevcuttur. Ara stok alanının sistemde bulunan parça sayısının bir eksiğinden daha az olması durumunda ise bu koşul çözüm yönteminin uygulanması aşamasında problem kısıtlarına dahil edilmelidir. Ara stok alanının sıfır olduğu durumda da aynısı geçerlidir.

Akış tipi bir sistemde ara stok alanının sınırsız olmadığı durumlarda makineler bloke olabilmektedir. Ara stok alanının dolu olması durumunda bir parçanın bir makine üzerindeki işlemi tamamlanınca bir sonraki makinede halen işlenmekte olan bir başka parça varsa, işlemi tamamlanan parça makine üzerinde beklemek zorunda kalacaktır. Bu durumda makine yeni bir parçanın işlemine başlayamayacağı için bloke olur. Ancak ara stok alanı sınırsız ise makinelerin bloke olması gibi bir durumla karşılaşmaz. Bu çalışmada da literatürdeki çok sayıda çalışmada olduğu gibi sınırsız ara stok alanı varsayımı yapılmıştır. Bu nedenle makinelerin bloke olması durumu ile karşılaşılmamaktadır.

Çalışmanın bu bölümündeki amaç her parça için esnek işlemin hangi makinede yapılacağına karar vererek son parçanın 3. makinedeki tamamlanma zamanı

değeri olarak tanımlanan yayılma zamanını (makespan) enküçükmektir. Bu amaç fonksiyonu çıktı miktarını enbüyüklemeyle eşdeğerdir. Sistemde özdeş parçaların üretiliyor olması nedeniyle amaç fonksiyonu olarak yayılma zamanının enküçüklenmesi daha uygun bulunmuştur.

Kesin çözüm yöntemi olarak geliştirilen matematiksel modelde aşağıdaki parametreler kullanılacaktır.

f^i : Makine i 'deki sabit işlem süresi ($i = 1, 2, 3$)

s : Esnek işlem süresi

Modelde yer alan karar değişkenleri ise şu şekildedir:

T_i^j : i parçasının j makinesinde işleme başlama zamanı.

C_{max} : n . parçanın üçüncü makinedeki tamamlanma zamanı.

$$x_i^j = \begin{cases} 1; & i \text{ işinin esnek işlemi } j \text{ makinesine atanmış ise} \\ 0; & \text{değilse} \end{cases}$$

Ele alınan problemin karma tamsayı modeli şu şekilde oluşturulmuştur.

$$\min \quad C_{max} \quad (3.1)$$

Öyle ki

$$C_{max} \geq T_n^3 + f^3 + s.x_n^3 \quad (3.2)$$

$$T_i^j \geq T_{i-1}^j + f^j + s.x_{i-1}^j \quad \forall i \neq 1, \forall j \quad (3.3)$$

$$T_i^j \geq T_i^{j-1} + f^{j-1} + s.x_i^{j-1} \quad \forall i, \forall j \neq 1 \quad (3.4)$$

$$x_i^1 + x_i^2 + x_i^3 = 1 \quad \forall i \quad (3.5)$$

$$T_1^1 \geq 0 \quad (3.6)$$

$$x_i^j \in \{0, 1\} \quad \forall i, \forall j \quad (3.7)$$

Modelde 3.1 numaralı denklem amaç fonksiyonunu belirtmektedir. Problemdaki amaç yayılma zamanını enküçükmektir. 3.2 numaralı eşitsizlik C_{max} değerinin

son parçanın son makine olan makine 3'teki tamamlanma zamanından büyük olması gerektiğini belirtmektedir. Son parçanın makine 3'teki tamamlanma zamanı o makinedeki işleme başlama zamanına makinenin sabit işlem süresi f^3 değerinin ve esnek işlem ataması yine makine 3'te ise s değerinin eklenmesiyle bulunmaktadır. 3.3 numaralı kısıt i parçasının j makinesinde işleme başlayabilmesi için $i - 1$ parçasının aynı makinede sabit işleminin ve o makineye atanmış ise esnek işleminin tamamlanmış olması gerektiğini belirtmektedir. Bir sonraki kısıt olan 3.4 kısıtı ise benzer durumu makineler için sağlamaktadır. i parçasının j makinesinde işleme başlayabilmesi için i parçasının bir önceki makinede sabit işleminin ve esnek işlem ataması o makinede ise esnek işleminin tamamlanmış olması gerekmektedir. 3.5 numaralı kısıt klasik bir atama kısıtıdır. Her parça için esnek işlemin sadece bir makineye atanmasını sağlamaktadır. 3.6 kısıtı ise çizelgenin başlama zamanını belirtir. 3.7 ise x_i^j karar değişkeninin 0 veya 1 değerini alabileceğini belirtmektedir.

Geliştirilen matematiksel model, Intel ® Core™ 2 Quad CPU Q9400 2,66 Ghz, 4 GB Ram, 64 bit bir bilgisayarda, GAMS 23.8.2 arayüzü ve CPLEX 12.2 çözücüsü kullanılarak çözdürülmüştür. Problem bu model ile test edilirken veri boyutu büyüdükçe özellikle parça sayısı artırıldığında, çözüm zamanının da çok arttığı gözlemlenmiştir. Bazı problemler için 24 saat içinde dahi sonuç alınamayan durumlarla bile karşılaşmıştır. Gerçek hayat problemlerinde, özellikle tek tip parçanın üretildiği sistemlerde üretilen parça sayısı genelde çok fazladır. Bu nedenle geliştirilen matematiksel modelin gerçek hayat problemlerinde etkin bir kullanımının olmayacağı açıktır. Bu problem için sonuca çok daha kısa sürede ulaşmayı amaçlayan çözüm yöntemleri geliştirilmiştir. Bu bölümün geri kalanında önce geliştirilen sezgisel çözüm yönteminden elde edilen sonuçlardan ve algoritmanın performansından bahsedilecektir. Sonrasında ise kesin çözüme çok daha kısa sürede ulaşmak için geliştirilen dinamik programlama algoritmasından bahsedilecektir.

3.2 Temel Sonuçlar

Bu bölümde sezgisel çözüm yöntemi ve dinamik programlama algoritmalarının geliştirilmesinde kullanılacak, optimal çözümün çeşitli özelliklerini belirten bazı

önteorem ve teoremler ispatlanacaktır. 3 makine problemi için optimal esnek işlem atamalarının bilindiği bazı özel durumlara da yine bu başlık altında değinilmektedir. Aşağıdaki önteorem sıralamadaki ilk işin esnek işleminin atamasını belirlemektedir.

Önteorem 3.1 *İlk işin esnek işleminin makine 3'te olduğu optimal bir çizelge vardır.*

İspat: İlk işin esnek işleminin her bir makineye atandığında bu parça için her üç makinede oluşacak tamamlanma zamanı değerleri ele alınacaktır. Bu değerlere Tablo 3.1'de yer verilmiştir. Bu tabloda C_i^j i parçasının j makinesindeki tamamlanma zamanını belirtmektedir.

Tablo 3.1: İlk Parça İçin Esnek İşlem Ataması Alternatifleri

	$x_1^1 = 1$	$x_1^2 = 1$	$x_1^3 = 1$
C_1^1	$f^1 + s$	f^1	f^1
C_1^2	$f^1 + f^2 + s$	$f^1 + f^2 + s$	$f^1 + f^2$
C_1^3	$f^1 + f^2 + f^3 + s$	$f^1 + f^2 + f^3 + s$	$f^1 + f^2 + f^3 + s$

Bu durumda makinelerde oluşan tamamlanma zamanı değerlerine dikkat edildiğinde makine 3'te oluşan tamamlanma zamanı değerinin atamaya bağlı kalmaksızın aynı olduğu görülmektedir. Makine 1 ve makine 2'de oluşan tamamlanma zamanı değerleri ise en küçük değerini esnek işlem atamasının makine 3'te olduğu durumda almaktadır. Çizelgenin geri kalanının ilk işin ataması üzerine herhangi bir etkisi yoktur. Sonuç olarak, her üç makinede de en düşük tamamlanma zamanı değerleri esnek işlemin üçüncü makineye atandığı alternatifte elde edilmektedir. \square

Muth [31] tarafından 1979 yılında ispat edilen ve Pinedo [36]'da yer alan ters problem özelliği ele alınan 3 makineli bu sistem için de geçerlidir. Bu problem için ters problem şu şekilde ifade edilebilmektedir.

Orjinal problem parametrelerinin f^1 , f^2 , f^3 , s ve n olduğu durumda ters problem şu şekilde tanımlanmıştır: üretilecek parça sayısı olan n aynı kalmakta,

makinelerin işlem süreleri ise $\hat{f}^1 = f^1$, $\hat{f}^2 = f^2$ ve $\hat{f}^3 = f^3$ olarak değişmektedir. Esnek işlem süresi her makinede eşit olduğu için ters problemde de $\hat{s} = s$ aynı kalmaktadır.

Önteorem 3.2 *Orijinal problemin ve ters problemin optimal yayılma zamanı değerleri aynıdır.*

Önteorem 3.1 ve ters problem özelliğinin verildiği Önteorem 3.2'nin doğal bir sonucu Önteorem 3.3'tür.

Önteorem 3.3 *Son işin esnek işleminin makine 1'de olduğu optimal bir çizelge vardır.*

Ters dönebilirlik özelliği ile asıl problem için ilk işin esnek işlem atamasının makine 3'e yapıldığı optimal bir çizelge olduğunu bilmek, aynı zamanda son işin esnek işlem atamasının makine 1'e yapıldığı optimal bir çizelge olduğunu da bilmek anlamına gelmektedir. Önteorem 3.4'te optimal çizelgelerle ilgili bir başka sonucu ispatlamaktadır.

Önteorem 3.4 *$f^1 = f^2 = f^3$ durumunda ikinci parçanın esnek işleminin makine 1'e atanmadığı optimal bir çizelge vardır.*

İspat: Bu önteoremin ispatı için ikinci işin esnek işleminin makine 1'e atandığı durum ile makine 2'ye atandığı durumda her üç makinede de ortaya çıkacak tamamlanma zamanı değerlerini ele almak yeterli olacaktır. Önteorem 3.1'e göre ilk işin esnek işleminin makine 3'e atandığı optimal bir çizelge olduğu bilinmektedir. Makine 2'ye yapılan atama sonucu üç makine için de elde edilen tamamlanma zamanı değerleri makine 1'e yapılan atama sonucu elde edilecek tamamlanma zamanı değerlerinden büyük değildir. Bu tamamlanma zamanı değerlerine Tablo 3.2'de yer verilmektedir.

Tablo 3.2'ye göre, ikinci işin esnek işlemi makine 2'ye atandığında her üç makinede ortaya çıkan tamamlanma zamanı değerleri, makine 1'e atandığından daha büyük

Tablo 3.2: İkinci Parçanın Esnek İşleminin Ataması

	$x_2^1 = 1$	$x_2^2 = 1$
C_2^1	$2f+s$	$2f$
C_2^2	$3f+s$	$3f+s$
C_2^3	$4f+s$	$4f+s$

olmaktadır. Bu da ikinci işin esnek işleminin ilk makineye atanmadığı optimal bir çizelge olduğunu ispatlamaktadır. \square

Teorem 3.1 her üç makinedeki sabit işlem zamanlarının aynı olduğu özel durumda optimal esnek işlem atamalarını vermektedir.

Teorem 3.1 $f^1 = f^2 = f^3$ ise, esnek işlem atamasının;

$$\begin{aligned} x_j^3 &= 1 & j=1,4,7,\dots,3k-2 & k=1,\dots,n \\ x_j^2 &= 1 & j=2,5,8,\dots,3k-1 & k=1,\dots,n \\ x_j^1 &= 1 & j=3,6,9,\dots,3k & k=1,\dots,n \end{aligned}$$

olduğu optimal bir çizelge vardır.

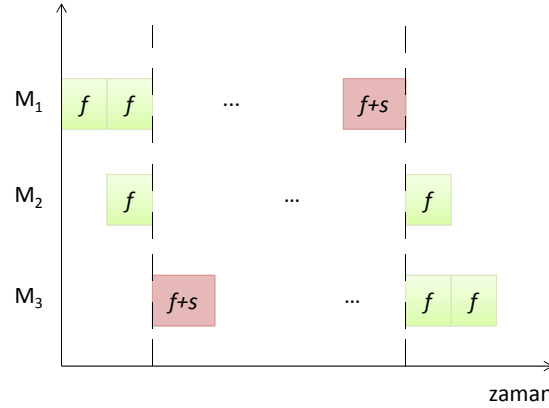
İspat: Bu teoremin ispatı için öncelikle optimal yayılma zamanı için bir alt sınır belirlenecek sonra da teoremden verilen atamanın bu alt sınıra eşit olduğu gösterilecektir. Üç makinenin ve tek bir esnek işlemin bulunduğu bu durum için bir alt sınır değeri bulunmak istenirse, çizelgenin üç parça halinde incelenmesi daha anlaşılır olmaktadır. Her bir parçanın minimize edilmeye çalışılması toplamda olabilecek en küçük tamamlanma zamanı değerini vermektedir. Çizelgenin başında ve sonunda oluşan engellenemeyen boş zamanlar ve arada kalan kısım olmak üzere çizelge üç bölüme ayrılabilir. Başlangıçta oluşan engellenemeyen boş zaman üçüncü makinenin ilk parçayı işlemeye başladığı ana kadarki bekleme süresidir yani ilk parçanın makine 1 ve makine 2'deki işleme sürelerinin toplamına eşittir. Bu sürenin Önteorem 3.1 ve Önteorem 3.4'ten $f + f = 2f$ olduğu bilinmektedir. Çizelgenin sonunda yer alan engellenemeyen boş zaman ise makine 1'in son parçayı işledikten sonra ne kadar boş beklediğidir. Bütün esnek işlemlerin arada kalan kısımda yani engellenemeyen boş zamanların haricindeki bölgede atandığı düşünülürse son parçanın makine 1'de işlenmesinden sonra makine 2 ve makine

3'te işleme süresi toplamı iki makinedeki sabit işlem süreleri toplamı olarak ve $2f$ şeklinde bulunmaktadır.

İspatı üç durum için incelemek daha açık ve anlaşılır olacaktır. Her bir durum için, alt sınır değerleri ve esnek işlem atamalarının ilk parçadan başlamak üzere son parçaya kadar sırasıyla makine 3, makine 2 ve makine 1'e yapılarak yani $3 - 2 - 1$ dizilimiyle elde edilecek C_{max} değerleri karşılaştırılmaktadır.

Durum 1: Parça sayısı $3n$ ise

Engellenemeyen boş zamanlar her üç durumun alt sınır değerini hesaplamak için de aynı kalmaktadır. Değişen, çizelgenin arada kalan kısmı olmaktadır. Bu alan Şekil 3.2'de görüldüğü üzere makine 3'ün ilk parçanın işlemine başlamasından makine 1'in son parçanın işlemini tamamlamasına kadar olan zaman dilimi olarak tanımlamak mümkündür.



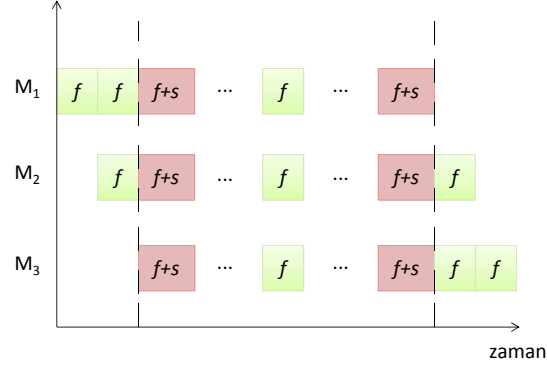
Şekil 3.2: Durum 1 İçin Alt Sınır Gantt Şeması

Arada kalan kısımda $3n - 2$ adet parçanın her bir makinedeki üç sabit işlemi ve $3n$ sayıda parçanın herhangi bir makinede esnek işlemi yapılmalıdır. Şekil 3.2'de görüldüğü gibi her makine için engellenemeyen boş zamanlar yani çizilen çizgilerin sağ ve sol tarafında kalan kısımlarda toplamda $2f$ vardır. Elde edilmek istenen alt sınır olduğundan hiç boş zaman oluşmadan bu çizelgenin tamamlandığı varsayılır. Yapılacak işlemler üç makineye eşit olarak dağıtılıp ve eş zamanlı tamamlandığı düşünüldüğünde durum 1 için alt sınır değeri aşağıdaki gibi elde edilebilir.

$$\text{Alt sınır}(1) = 4f + [3(3n - 2)f + (3n)s]/3 = (3n + 2)f + ns$$

Bu değerin elde edilmesinde $4f$ değeri engellenemeyen boş zamanlardan gelen toplam değerdir. f değerinin $3n - 2$ ile çarpılma sebebi ise her makine için $2f$ değeri çizgilerin dışında kaldığından, aradaki bölgede çizelgelenecek olan toplam $3n - 2$ parça kalmış olmasındandır. Alt sınır değeri elde edilmek istendiği için arada kalan bölgedeki toplam sabit ve esnek işlem zamanlarının tam 3'e bölünerek makinelerin dengelendiği varsayılmaktadır. Böylece bu durum için elde edilen Alt sınır (1) değeri $= (3n + 2)f + ns$ olarak elde edilir.

Durum 1 için $3 - 2 - 1$ esnek işlem atama dizilimi ile elde edilen yayılma zamanı değerini ise $C_{max}(1)$ olarak gösterelim. Parça sayısı $3n$ olduğundan her makineye eşit sayıda esnek işlem ataması yapılacaktır. Esnek işlemlerin ataması dikkate alındığında ilk parçanın esnek işleminin makine 3'te, son parçanın esnek işleminin ise makine 1'de olduğu görülmektedir. Şekil 3.3'te bu atamanın Gantt Şemasına yer verilmektedir.



Şekil 3.3: $C_{max}(1)$ Gantt Şeması

Kurala göre her makineye n adet esnek işlem atanacağından $C_{max}(1)$ değeri şöyle bulunmaktadır.

$$C_{max}(1) = 2f + (3n - 2)f + ns + 2f = (3n + 2)f + ns$$

Değer hesaplanırken eklenen 2 adet $2f$ değeri Şekil 3.3'de de görüldüğü üzere makine 1 ve makine 3'te oluşan engellenemeyen boş zamanlardır. Geriye $3n - 2$ adet parçanın sabit işlemi ve n parçanın esnek işlemi toplamı kalır. Engellenemeyen boş zamanlar ile oluşan bu işlem süreleri toplamı $C_{max}(1)$ değerini vermektedir.

Alt sınır(1) $= C_{max}(1)$ olarak bulunduğuna göre parça sayısı $3n$ olduğu

durumda $3 - 2 - 1$ esnek işlem ataması dizilimiyle elde edilen çizelge optimaldir.

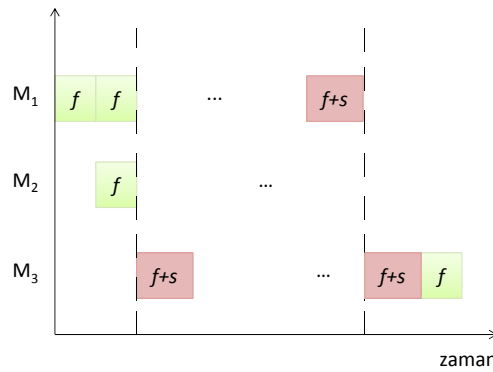
Durum 2: Parça sayısı $3n - 1$ ise

Parça sayısı 3'ün tam katı olmadığı için esnek işlem atamaları makinelere eşit olarak dağıtılamayacaktır. Bir makineye atanan toplam esnek işlem sayısı diğer iki makinedekinden 1 eksik olmak zorundadır. Bu durumda makinelere atanacak esnek işlem sayıları n , n ve $n - 1$ adet olacaktır. Alt sınır bulunmaya çalışıldığı için n tane esnek işlemin atandığı makine dikkate alınır. Alt sınır(2) değeri aşağıdaki gibi hesaplanır.

$$\text{Alt sınır}(2) = 4f + s + [3(3n - 3)f + (3n - 3)s]/3 = 4f + s + (3n - 3)f + (n - 1)s/3$$

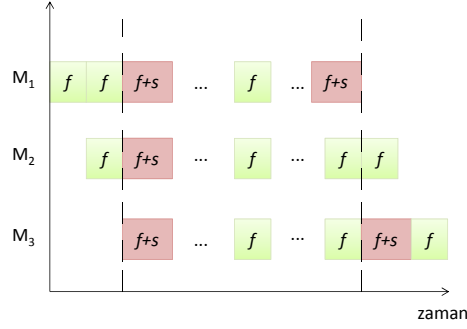
$$\text{Alt sınır}(2) = 4f + s + (3n - 3)f + (n - 1)s = (3n + 1)f + ns$$

Alt sınır(2)'nin elde edilmesini daha açık görebilmek için Şekil 3.4 incelenebilir. Çizgilerin sağında ve solunda kalan alanlar bu durum için oluşan engellenemeyen boş zamanları temsil etmektedir. Parça sayısı 3'ün tam katı olmadığı için bu durumda esnek işlem atamaları son makineyle başlayıp ikinci makine ile bitecektir. Öntem 3.3'te son işin esnek işlem atamasının her zaman ilk makinede olduğu optimal bir çizelge olduğu gösterilmiştir. Bu durumda son parçanın esnek işlemi makine 1'e atanmış gibi gösterilecektir.



Şekil 3.4: Durum 2 İçin Alt Sınır Gantt Şeması

$C_{max}(2)$ değeri içinse $3 - 2 - 1$ esnek işlem ataması dizilimi ile elde edilecek değer şöyle belirlenmektedir. Parça sayısı $3n - 1$ olduğuna göre bu diziye uygun atama yapıldığında son parçanın esnek işlemi makine 2'ye



Şekil 3.5: $C_{max}(2)$ İçin Gantt Şeması

atanmaktadır. Bu atamanın nasıl yapıldığına Şekil 3.5'te yer verilmektedir.

Buna göre $C_{max}(2)$ değeri şöyle bulunur: n parçanın esnek işlemi makine 3'e, n parçanın esnek işlemi makine 2'ye ve $n - 1$ parçanın esnek işlemi ise makine 1'e atanır. Arada kalan bölgenin makine 1'in tamamlanması ile bittiği bilindiğinden makine 1'e atanan esnek işlem sayısı $C_{max}(2)$ değerini hesaplariken belirleyici olacaktır.

$$C_{max}(2) = 2f + (3n - 3)f + (n - 1)s + 2f + s = (3n + 1)f + ns$$

Alt sınır(2) = $C_{max}(2)$ olarak bulunduğuna göre parça sayısı $3n - 1$ olduğu durumda 3 - 2 - 1 dizilimiyle yapılacak atamayla elde edilen çizelge optimaldir.

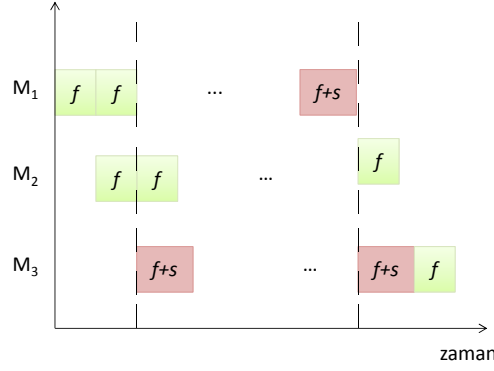
Durum 3: Parça sayısı $3n - 2$ ise

Parça sayısı 3'ün tam katı olmadığı için Durum 2'dekine benzer bir şekilde esnek işlem ataması bütün makinelere eşit sayıda dağıtılamamaktadır. $3n - 2$ adet parça olduğuna göre alt sınır belirlenirken esnek işlemler makinelere $n - 1$, $n - 1$ ve n adet olacak şekilde dağıtılır. Tamamlanma zamanı hesaplanacağından n adet esnek işlemin atandığı makine dikkate alınacaktır. Alt sınır(3) şu şekilde hesaplanır.

$$\text{Alt sınır}(3) = 4f + s + [3(3n - 4)f + (3n - 3)s]/3$$

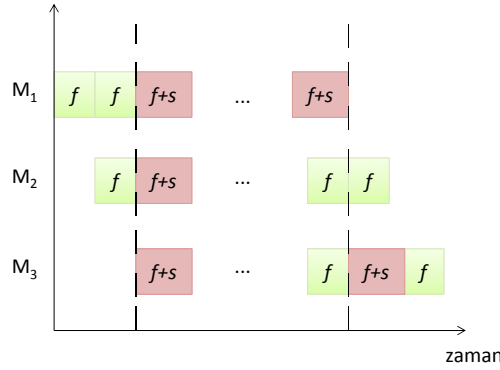
$$\text{Alt sınır}(3) = 4f + (3n - 2)f + ns = 3nf + ns$$

Alt sınır (3) için, engellenemeyen boş zamanlara ve çizelgenin arada kalan kısmına Şekil 3.6'da yer verilmektedir.



Şekil 3.6: Durum 3 İçin Alt Sınır Gantt Şeması

3 – 2 – 1 dizilimyle yapılacak atamanın C_{max} değeri bulunmak istendiğinde ise parça sayısı $3n - 2$ olduğundan son parçanın esnek işlemi bu atama yöntemine göre makine 3'te olmaktadır. Bu durumda atamanın oluşturacağı Gantt Şeması Şekil 3.7'de verildiği gibi olacaktır.



Şekil 3.7: $C_{max}(3)$ İçin Gantt Şeması

Buna göre $C_{max}(3)$ değeri şöyle bulunur: n parçanın esnek işlemi makine 3'e, $n - 1$ parçanın esnek işlemi makine 2'ye ve $n - 1$ parçanın esnek işlemi ise makine 1'e atanmaktadır. Arada kalan bölgenin makine 1'in tamamlanması ile bittiği bilindiğinden makine 1'e atanan esnek işlem sayısı $C_{max}(3)$ değerini hesaplarken belirleyici olacaktır.

$$C_{max}(3) = 2f + (3n - 4)f + (n - 1)s + 2f + s = 3nf + ns$$

Alt sınır(3) = $C_{max}(3)$ olarak bulunduğuna göre parça sayısı $3n - 2$ olduğu durumda 3 – 2 – 1 dizilimiyle yapılacak atamayla elde edilen çizelge optimaldir.

3 – 2 – 1 diziliminde karşılaşılabilecek her üç durum da incelenerek

makinelardaki işlem sürelerinin eşit olduğu durum için bu dizilimin optimal sonuç vereceği ispat edilmiştir. \square

Aşağıdaki teoremde bir başka özel durum için optimal çözüm ispatlanacaktır.

Teorem 3.2 *Herhangi bir k ve j makinesi için ($k, j = 1, 2, 3, k \neq j$), $f^k + s \leq f^j$, ise her parça için esnek işlem atamalarının k makinesine yapıldığı bir optimal çizelge vardır.*

İspat: Bu durumun geçerli olduğu işlem sürelerine sahip bir problemde esnek işlemler hep k makinesine atanacaktır. Çünkü k makinesinin sabit işlem süresine s esnek işlem süresi eklenmiş hali dahi j makinesinin sabit işlem süresinden küçük ya da eşittir. Böyle bir durumda, sistemdeki üçüncü makine l makinesi olsun.

- Eğer $f^l \leq f^j$ ise yayılma zamanını belirleyen j makinesi olacaktır. Bu durumda optimal yayılma zamanı $C_{max} = f^k + f^l + n(f^j)$ olacaktır.
- Eğer $f^j \leq f^l$ ise yayılma zamanını belirleyen l makinesi olacaktır. Bu durumda optimal yayılma zamanı $C_{max} = f^k + f^j + n(f^l)$ olacaktır.

\square

Bundan sonraki bölümlerde, bu bölümde önteorem ve teoremlerle verilen optimal çözüm özellikleri dikkate alınarak geliştirilen sezgisel yöntem ve dinamik programlama algoritması anlatılmaktadır.

3.3 Sezgisel Çözüm Yöntemi

Bu bölümde, 3 makineli akış tipi probleminin çözümü için ilk olarak başvurulan yöntem olan sezgisel algoritma adımları, algoritmanın test edilmesi için yapılan deneysel çalışma ve algoritmanın performansına değinilmektedir.

Sezgisel yöntem geliştirilirken önceki bölümde anlatılan problemin optimal çözüm özellikleri dikkate alınmaktadır. Öntelem 3.1 ve 3.3 ile sezgisel yöntemde, hem ilk hem de son iş için herhangi bir hesaplama yapılmasına gerek kalmaksızın işlerin atamaları yapılmaktadır. Bu da çözüm süresini kısaltarak yöntemin etkinliğinin artmasına yardımcı olmaktadır. Teorem 3.1 ve 3.2'e dayanarak problem verileri türetilmiştir. Optimal sonucu zaten bilinen problem verileri dikkate alınmayarak sonucu bilinmeyen problem veri setleri için sezgisel yöntem test edilmiştir.

Geliştirilen bu algoritma Crama ve Gültekin [12] tarafından iki makineli tek esnek işlemin yer aldığı problem için geliştirilen “*iki parça ileri (2PA)*” kesin çözüm algoritmasını temel almaktadır. Algoritmanın mantığı benzer şekildedir. Ancak 2 makineli sistemleri ele alan Crama ve Gültekin [12]'in algoritmasında iki parça sonrası için ikinci makinede boş zaman oluşup oluşmayacağı kontrolü yapılırken bu çalışmadaki algorithmada çizelgede hem bir parça sonrası hem de iki parça sonrası kontrol edilmektedir. Bu algoritma 3 makineli bir sistem için geliştirildiğinden iki farklı karşılaştırma yapılmaktadır. Önce makine 1 ve makine 2 için olası tamamlanma zamanları karşılaştırılıp daha sonra aynı karşılaştırma makine 2 ve makine 3 için yapılmaktadır.

3.3.1 Algoritma adımları

Geliştirilen sezgisel algoritmanın adımları aşağıdaki gibi ifade edilebilir.

Adım 1: Öntelem 3.1'e dayanarak ilk parçanın esnek işlemi makine 3'e atanır.

Adım 2: İkinci parçadan itibaren sırasıyla şu kontroller yapılarak atamaya karar verilir.

- Eğer bu parçanın esnek işleminin makine 1'e atanması, makine 2'de boş zaman oluşmasına sebep olmuyorsa, bir sonraki parça için kontrol edilir. Sonraki parça için de boş zaman oluşmuyorsa atama makine 1'e yapılır.
- İlk şart sağlanmıyorsa, bu parçanın esnek işlemi makine 2'ye atanırsa makine 3'te boş zamana sebep olup olmadığı kontrol edilir, boş zaman olmuyorsa bir sonraki parça için kontrol yapılır, yine boş zaman oluşmuyorsa atama makine 2'ye yapılır.

- Bu iki şart da sağlanamıyorsa atama makine 3'e yapılır.

Adım 3: $n - 1$. parçaya kadar adım 2 tekrar edilir.

Adım 4: Son parçanın ataması ilk makineye yapılır.

Geliştirilen polinom zamanlı algoritmanın ayrıntılı olarak adımlarına Algoritma 1 olarak yer verilmiştir.

Algoritma 1:

Girdi: f^1, f^2, f^3, s, n

Çıktı: C_i^j, C_{max}, x_i^j

1. İlk işin esnek işlemini üçüncü makineye ata.
 2. Her makine için ilk işin tamamlanma zamanını hesapla.
 3. $i = 2$ 'den $(n - 1)$ ' e kadar **yap**.
 4. **eğer** $C_i^1 + 2f^1 + s \leq C_i^2 + f^2$ ve $C_i^1 + f^1 + s \leq C_i^2$ ise
 5. i . işin esnek işlemini makine 1'e ata.
 6. i . işin her üç makinedeki tamamlanma zamanlarını hesapla.
 7. **değilse**
 8. **eğer** $\max((C_i^1 + f^1), C_i^2) + 2f^2 + s \leq C_i^3 + f^3$ ve $\max((C_i^1 + f^1), C_i^2) + f^2 + s \leq C_i^3$ ise
 9. i . işin esnek işlemini makine 2'ye ata.
 10. i . işin her üç makinedeki tamamlanma zamanlarını hesapla.
 11. **değilse**
 12. i . işin esnek işlemini makine 3'e ata.
 13. i . işin her üç makinedeki tamamlanma zamanlarını hesapla.
 14. **ise sonlandır**.
 15. **ise sonlandır**.
 16. n . işin esnek işlemini ilk makineye ata.
 17. n . işin her üç makinedeki tamamlanma zamanlarını hesapla.
 18. n . işin makine 3'teki tamamlanma zamanını C_{max} olarak belirle.
 19. **döngü sonlandır**.
-

Bir örnekle algoritmanın işleyişini daha iyi anlamak mümkündür.

Örnek 1: 5 parçanın üretildiği bir sistemde, sabit işlem süreleri sırasıyla 9, 8, 11 ve esnek işlem süresi 7 olsun.

Çözüm:

- İlk işin esnek işlemi son makineye atanır. Üç makine için de tamamlanma zamanları hesaplanır.

$$C_1^1 = f^1 \rightarrow 9$$

$$C_1^2 = f^1 + f^2 \rightarrow 9 + 8 = 17$$

$$C_1^3 = f^1 + f^2 + f^3 + s \rightarrow 9 + 8 + 11 + 7 = 35$$

Adım 1: İkinci iş için algoritmada belirlenen makine 1 ve makine 2 karşılaştırması yapılır.

$C_1^1 + 2 \cdot f^1 + s \leq C_1^2 + f^2$ ve $C_1^1 + f^1 + s \leq C_1^2$ şartlarının sağlanıp sağlanmadığı kontrol edilir.

$32 \not\leq 24$ ve $25 \not\leq 17$ şartlardan birinin sağlanmaması atamanın yapılmaması için yeterlidir ki bu durumda her iki şart da sağlanmamaktadır. Atama makine 1'e yapılmaz, bir sonraki adıma geçilir.

Adım 2: İkinci iş için algoritmada belirlenen makine 2 ve makine 3 karşılaştırması yapılır.

$\max\{C_1^1 + f^1, C_1^2\} + 2f^2 + s \leq C_1^3 + f^3$ ve $\max\{C_1^1 + f^1, C_1^2\} + f^2 + s \leq C_1^3$ şartlarının sağlanıp sağlanmadığı kontrol edilir.

$41 \leq 46$ ve $33 \leq 35$ Her iki şartta sağlandığı için ikinci parçanın esnek işlem ataması makine 2'ye yapılır. İkinci parçanın tamamlanma zamanları hesaplanır.

$$C_2^1 = C_1^1 + f^1 \rightarrow 9 + 9 = 18$$

$$\max\{C_2^1, C_1^2\} + f^2 + s \rightarrow \max\{18, 17\} + 8 + 7 = 33$$

$$\max\{C_2^1, C_1^3\} + f^3 \rightarrow \max\{33, 35\} + 11 = 46$$

Adım 3: Adım 2, üçüncü ve dördüncü parça için tekrar edilir, oluşan tamamlanma zamanları hesaplanır.

$$C_3^1 = 27 \quad C_4^1 = 36$$

$$C_3^2 = 41 \quad C_4^2 = 56$$

$$C_3^3 = 64 \quad C_4^3 = 75$$

Adım 4: Son parçanın esnek işlem ataması makine 1'e yapılır. Yayılma zamanı C_{max} hesaplanır.

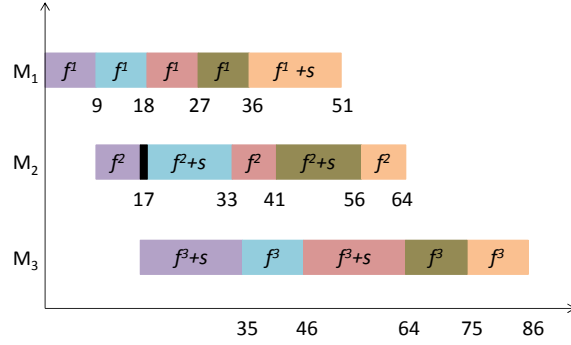
$$C_5^1 = C_4^1 + f^1 + s \rightarrow 36 + 9 + 7 = 51$$

$$C_5^2 = \max\{C_4^2, C_5^1\} + f^2 \rightarrow \max\{56, 51\} + 8 = 64$$

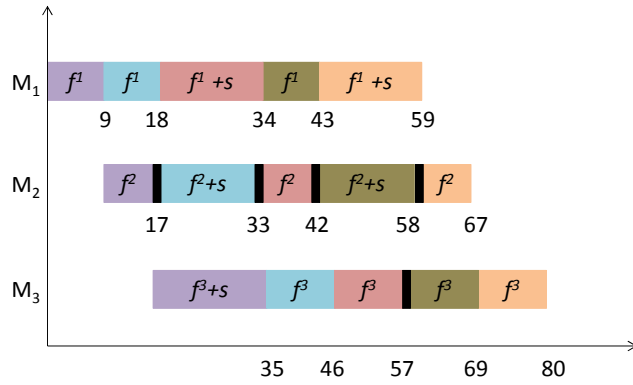
$$C_5^3 = \max\{C_4^3, C_5^2\} + f^3 \rightarrow \max\{75, 64\} + 11 = 86$$

$C_{max}(sezgisel) = 86$ olarak bulunur. Matematiksel model ile elde edilen optimal sonuç ise $C_{max}(optimal) = 80$ olarak bulunmuştur.

Şekil 3.8'de Örnek 1 için sezgisel algoritma ile elde edilen çizelgeye, Şekil 3.9'da ise optimal çizelgeye yer verilmiştir. Şekilde yer alan siyah kutucuklar boş zamanları temsil etmektedir. Optimal çizelgede, daha fazla boş zaman oluşmuş olmasına rağmen yayılma zamanı daha düşüktür.



Şekil 3.8: Örnek 1 İçin Sezgisel Yöntemle Bulunan Çizelge



Şekil 3.9: Örnek 1 İçin Optimal Çizelge

3.3.2 Deneysel çalışma

Bu bölümde, geliştirilen algoritmanın performansının test edilmesi için yapılan deneysel çalışma anlatılmaktadır. Deneysel çalışmada problemin çözüm süresini

etkileyen parametreler belirlenerek bunların deęiştirilmesiyle algoritmanın verdięi sonuçların nasıl deęiştii gözlemlenmektedir.

Bu problem analizi için beş faktörün yer aldığı bir deney tasarımı yapılmıştır. Faktörlerden 4 tanesi (f^1, f^2, f^3 ve s) iki farklı seviyede değerlendirilirken iş sayısını temsil eden n için ise üç farklı seviye ele alınmıştır. Faktörlerin ve seviyelerinin hangi deęerleri aldığına Tablo 3.3'te yer verilmektedir.

Tablo 3.3: Deney Faktörleri

Faktör	Tanımı	Seviye 1	Seviye 2	Seviye 3
n	İş sayısı	20	50	100
f^1	Makine 1'deki işlem süresi	D	Y	
f^2	Makine 2'deki işlem süresi	D	Y	
f^3	Makine 3'teki işlem süresi	D	Y	
s	Esnek işlem süresi	D	Y	

Problemin test edilmesi aşamasında her n deęeri için $2^4 \times 3 = 48$ farklı kombinasyon ve her birisinden 5 tekrar olmak üzere toplamda 240 farklı problem çözdürülmüştür. Makinelerdeki işlem zamanları belirlenirken her bir parametrenin türetilceęi aralıkları oluşturmak için iki farklı alt taban deęeri ve bu taban deęere eklenecek bir sabit deęer belirlenmiştir. Bu iki taban deęeri faktör dizaynında yer alan sırasıyla D ve Y'nin temsil ettięi 10 ve 50 olarak belirlenmiştir. Eklenen sabit sayı ise 30 olarak seçilmiştir.

Parametrelerin türetilmesi esnasında öncelikle f^1 ve s deęerleri herhangi bir kısıtlama olmadan seçilen taban deęerine belirlenen sabit deęeri eklenerek elde edilen aralıktan, düzgün dağılıma göre türetilmiştir. f^2 ve f^3 parametrelerinin deęerleri belirlenirken, Öntoorem 3.2'de bahsedilen kolay durumları içermeyecek şekilde deęerler almaları sağlanmıştır. Bu nedenle f^2 ve f^3 deęerlerinin türetilcekleri aralıklar belirlenirken sadece önceden verilen taban ve sabit deęerler kullanılmamıştır. Kolay durum içermeyecek şekilde alt ve üst sınır belirlenerek f^2, f^3 parametrelerinin alacağı deęerler bu sınırlar içinden düzgün dağılıma göre türetilmiştir. f^2 deęeri belirlenirken alt ve üst sınır deęeri aşağıda yer alan eşitlikler yardımıyla türetilmiştir.

Verilerin türetilmesi sırasında kullanılan notasyon şu şekilde açıklanabilir:

f_{taban}^j : Değerin türetileceği seviye için belirlenen taban değer.

f_{tavan}^j : Değerin türetileceği seviye için belirlenen tavan değer.

f_{alt}^j : Bulunduğu seviyedeki taban değer ile kolay durum oluşturmayacak şekilde bir önceki belirlenen parametre değerlerine bağlı olarak alabileceği en küçük değer büyük olanı.

$f_{üst}^j$: Bulunduğu seviyedeki tavan değer ile kolay durum oluşturmayacak şekilde bir önceki belirlenen parametre değerlerine bağlı olarak alabileceği en büyük değer küçük olanı.

Bu durumda sırasıyla f_{alt}^2 , $f_{üst}^2$, f_{alt}^3 ve $f_{üst}^3$ değerlerinin türetileceği aralıklar şu şekilde olacaktır.

$$f_{alt}^2 = \max\{f_{taban}^2, f^1 - s\}$$

$$f_{üst}^2 = \min\{f_{tavan}^2, f^1 + s\}$$

f^3 parametresi belirlenirken f^1 ve s 'in yanı sıra f^2 değerinin ne olduğu da önem kazanmıştır ve değerlerin türetileceği eşitlikler şu şekilde belirlenmiştir.

$$f_{alt}^3 = \max\{f_{taban}^3, f^1 - s, f^2 - s\}$$

$$f_{üst}^3 = \min\{f_{tavan}^3, f^1 + s, f^2 + s\}$$

Bütün parametrelerin düzgün dağılımla türetileceği aralıklar her kombinasyonda ve her problemde değişerek bu şekilde elde edilmiştir.

Matematiksel modelin sonuçları GAMS 23.8.2 arayüzü kullanılarak CPLEX 12.2 çözücüsü yardımıyla elde edilmiştir. Sezgisel algoritma ise Dev C++ ile kodlanmıştır. Analizlerin yapılmasında kullanılan bilgisayar Intel ® Core™ 2 Quad CPU Q9400 2,66 Ghz, 4 GB Ram özelliklerine sahiptir.

Belirtilmesi gereken bir başka nokta da matematiksel modelin çalıştırılmasında uzun süre sonuç alınamayan problemlerle karşılaşma ihtimaline karşı her bir problem için 900 saniye zaman limiti konmuştur. Bu süre zarfında optimal sonuç elde edilememiş ise elde edilen değer ve bu değer alt sınıra olan

uzaklığını gösteren yüzde boşluk değer çözdürücünün hatası olarak sonuçlara dahil edilmiştir. Problemler için 900 saniye gibi bir zaman limitinin tercih edilme sebebi gerçek hayatta bu problemin operasyonel bir karar olarak karşımıza çıkacak olmasıdır. Bir üretim sisteminde böyle bir problemin, üretimin nasıl yapılacağına karar vereceği göz önünde bulundurulduğunda günde en az bir defa hatta gün içinde birkaç kez çözdürülmesi gerekeceği düşünülebilir. Üretim için 15 dakikalık bir aksama ciddi maliyetlere neden olabileceği için en fazla bu kadar süre içinde bir çözümün elimizde olması istenmiştir.

Deneysel çalışmanın tamamını içinde barındıran özet bilgilere Tablo 3.4'te yer verilmektedir. Bu tabloda *Gams Boşluk* olarak ifade edilen değerler 900 saniyelik zaman limitinde optimal sonuca ulaşamadığı takdirde şu şekilde hesaplanmaktadır:

$$\text{Gams Boşluk(\%)} = (\text{Mevcut Çözüm} - \text{Alt sınır Değeri}) / \text{Alt sınır Değeri} \times 100$$

Sezgisel yöntemin verdiği sonucun performansını belirten *Sezgisel Hata* ise şu şekilde hesaplanmaktadır:

$$\text{Sezgisel Hata(\%)} = (\text{Sezgisel Çözüm} - \text{Optimal (Alt sınır) Değer}) / \text{Optimal (Alt sınır) Değer} \times 100$$

Hesaplama hem optimal hem alt sınır değerinin yer alma sebebi eğer söz konusu problem için çözdürücü vasıtasıyla optimal sonuca ulaşılabilmişse bu değer optimalden olan uzaklığı, eğer ulaşılammış ve çözdürücünün de belli bir hatası varsa sezgisel yöntemin alt sınırdan uzaklığı dikkate alınmıştır.

Tablo 3.4: Bütün Problemleri İçeren Genel Tablo

Parça Sayısı	GAMS Çözümleri			Ortalama	CPU	CPU
	Optimal çözülen problem sayısı	Zaman limitine takılan problem sayısı	Ortalama boşluk (%)	Sezgisel Hata(%)	GAMS (sn)	Sezgisel (sn)
20	79	0	0	0.93	3.76	0.02
50	56	23	0.24	0.63	259.77	0.05
100	46	33	0.24	0.47	372.72	0.04

Analizde her bir n deęeri için 80 problem çözdürüldüęü söylenmiřti ancak tabloda bütün n deęerleri için 79 adet örnek bulunmaktadır. Bunun sebebi her bir n deęerinde farklı parametre kombinasyonlarında olmak üzere, řartlara uygun (kolay durum içermeyen) veri türetilmemesinden kaynaklanmaktadır.

Tablodaki deęerlerle n sayısı arttıkça matematiksel modelle optimal sonuç alınan problem sayısının azaldığı görölmektedir. 20 parça varken bütün problemler için optimal sonuç elde edilmiř ancak parça sayısının 50 ve 100 olduęu durumlarda sırasıyla 23 ve 33 adet problemde zaman limitinden dolayı optimal sonuç elde edilememiřtir. Optimal sonucun bulunamadığı durumlarda bu örneklerin yüzde boşluk deęerlerinin ortalaması alınarak hata sütununa yazılmıřtır.

Sezgisel ve GAMS çözümleri karşılaştırıldığında her durumda sezgiselin çok daha kısa sürede sonuç verdięi görölmektedir. 50 ve 100 parça sayısında GAMS çözümlerinin 20 parçaya göre bir anda artma sebebi zaman limitine ulaşan problemlerin de ortalamaya dahil olmasıdır. 20 parçanın olduęu 79 problemde zaman limitine ulaşan problem olmamıřtır. Ancak sezgisel algoritma sonuçları, matematiksel modelle optimal sonucun elde edildięi sürelerle göre çok daha kısa sürede elde etmiřtir. Ayrıca n arttıkça sezgiselin hata payının azaldığı görölmüřtür. Problem boyutu büyüdükçe sezgisel algoritma optimale daha yakın sonuçlar verebilmektedir. Bu sonuca řöyle bir yorum getirilmiřtir. Algoritmanın iřleyiřinde bařlangıçta yapılabilecek optimalden farklı bir atamanın, fazla parçanın olduęu problemlerde, çizelgenin ilerleyen kısımlarında daha kolay telafi edilebileceęi düşünölmüř ve bu nedenle parça sayısı arttıkça optimale yakın sonuçlar bulunduęu yönünde yorum yapılmıřtır. Özdeş parçaların üretildięi bu sistemde parça sayısının çok daha büyük olacaęı düşünöldüğünde bu istenen bir sonuçtur. Çözüm süresi içinse böyle bir gözlem yapılamaz çünkü n deęerinin deęiřimini sezgiselin çözüm süresini kayda deęer bir oranda etkilememektedir. Sezgisel algoritmanın parça sayısına duyarlı olmaması da güçlü bir yönü olarak deęerlendirilebilmektedir.

n deęerlerinin her biri için ayrı ayrı tablolara da yer verilmiřtir. Bu tablolarda her bir faktör kombinasyonuna ait sonuçlara yer verilerek daha ayrıntılı bir analiz yapılmıřtır. Bu tablolarda faktörlerin seviyeleri belirtilmiř ve her faktör kombinasyonu için yapılan beř tekrarın ortalaması alınarak bu deęerler yazılmıřtır.

Tablo 3.5: 20 Parça İçin Özet Tablo

n	Faktörler				GAMS Boşluk (%)	Sezgisel Hata (%)	CPU GAMS (sn)	CPU Sezgisel (sn)
	f^1	f^2	f^3	s				
	Y	Y	Y	Y	0	0.46	0.81	0.02
	Y	Y	Y	D	0	3.62	6.22	0.02
	Y	Y	D	Y	0	0	0.05	0.03
	Y	Y	D	D	0	1.59	0.11	0.02
	Y	D	Y	Y	0	0.04	0.57	0.02
	Y	D	Y	D	0	1.44	0.34	0.02
	Y	D	D	Y	0	0.01	0.21	0.02
20	Y	D	D	D	0	2.11	3.14	0.02
	D	Y	Y	Y	0	0	0.05	0.02
	D	Y	Y	D	0	0.27	0.07	0.02
	D	Y	D	Y	0	0.81	0.16	0.02
	D	Y	D	D	0	1.94	0.49	0.02
	D	D	Y	Y	0	0.27	0.04	0.03
	D	D	Y	D	0	1.25	0.17	0.02
	D	D	D	Y	0	0.88	22.34	0.02
	D	D	D	D	0	0.27	26.47	0.02
Ortalama					0	0.93	3.76	0.02

Tablo 3.5'te yer alan değerler 20 parçalık problemlerin parametrelerinin 16 farklı kombinasyonunun her birinden yapılan beş tekrarın ortalamalarıdır. 20 parçalık örneklerde GAMS ile elde edilen bütün çözümler optimal değerlerdir. Sezgisel hatalar hesaplanırken sadece optimal değerlerden olan uzaklık formülü kullanılmıştır. GAMS çözüm süresinde kombinasyonlar arasında büyük değişiklikler görülebilmektedir. Sezgisel çözüm süresine ait değerler ise birbirine çok yakın değerler olmuşlardır. Sezgisel algoritma iki parametre kombinasyonuna ait bütün tekrarlarda optimal sonucu bulabilmiştir. Bu kombinasyonlar f^1, f^2, s değerinin büyük, f^3 değerinin küçük değer aldığı durum ve f^2, f^3, s parametrelerinin büyük, f^1 parametresinin ise küçük değer aldığı aralıktan türetilmiştir. Bu iki durumun ortak noktası ise bir sabit işlem süresi değeri ve s değerinin büyük değerler alacağı aralıktan türetilmesi ve ilk ya da son makinenin ise küçük değerler almasıdır.

Tablo 3.6: 50 Parça İçin Özet Tablo

n	Faktörler				GAMS	Sezgisel	CPU	CPU
					Boşluk (%)	Hata (%)	GAMS (sn)	Sezgisel (sn)
	f^1	f^2	f^3	s				
	Y	Y	Y	Y	0.21	1.16	360.27	0.05
	Y	Y	Y	D	3.33	1.23	720.21	0.06
	Y	Y	D	Y	0	0	0.19	0.07
	Y	Y	D	D	0.5	0.38	180.16	0.07
	Y	D	Y	Y	0	0.09	0.1	0.07
	Y	D	Y	D	0	0.26	0.17	0.06
	Y	D	D	Y	0	0.1	0.17	0.05
50	Y	D	D	D	1.8	0.83	546.09	0.05
	D	Y	Y	Y	0	0.1	0.1	0.05
	D	Y	Y	D	0	0.7	0.41	0.05
	D	Y	D	Y	0	0.21	2.76	0.05
	D	Y	D	D	2.07	1.61	900.1	0.05
	D	D	Y	Y	0.55	1.05	364.78	0.05
	D	D	Y	D	0.65	1.39	360.14	0.05
	D	D	D	Y	0	0.29	0.37	0.05
	D	D	D	D	1.7	0.74	720.23	0.05
Ortalama					0.68	0.63	259.77	0.055

Tablo 3.6'da ise 50 parça için kombinasyonların ortalamaları verilmiştir. Bu örneklerde GAMS için de hata değerinin bulunduğu görülmektedir. Bu hata değeri

GAMS ile zaman limitine ulařtıđında bulunmuř olan sonucun alt sınıra olan yüzde uzaklıđını göstermektedir.

Zaman limitine ulařan problemler için sezgiselin hatası hesaplanırken GAMS ile bulunan alt sınır kullanılmıřtır. 50 parçanın bulunduđu bu durumda sezgiselin bütün tekrarlarının optimal sonucu verdiđi tek bir kombinasyona ulařılmıřtır. Bu kombinasyon yine 20 parçalık örneklerde de optimal sonuca ulařılan faktör kombinasyonlardan biridir.

GAMS ile elde edilen sonuçlarda çok küçük uzaklıklarda deđerler bulunurken sezgisel ile elde edilen sonuçlarda da elde edilen deđerler optimalden ya da alt sınır deđerinden binde oranında uzaklıklardadır. Bu da kabul edilebilir bir oran olarak bahsedilebilir.

Çözüm süreleri karşılařtırıldıđında ise GAMS çözüm süresinin oldukça uzun olduđu görölmektedir. Ortalama deđerlerine bakıldıđında çözüm süresi olarak sezgiselin büyük bir üstünlüđu olduđu gerçektir. Sezgisel algoritma ile elde edilen çözümler ortalama olarak saniyenin 1/18'i gibi bir deđerdir.

Sezgiselin hata oranlarının büyük olduđu yerler incelendiđinde bu iki n deđerini için ortak kombinasyonların olduđu yerler gözlemlenmemiřtir.

100 parçanın olduđu örneklerin yer aldıđı Tablo 3.7'de yine her kombinasyon tekrarlarının ortalamaları bulunmaktadır. GAMS boşluk oranı bu tabloda daha büyük deđerler almaya başlamıřtır. Problem boyutunun büyümesiyle optimal sonuca ulařılamayıp zaman limitiyle duran problem sayısı artmıřtır. Bu tabloda bir diđer dikkat çeken husus ise sezgisel algoritmanın her tekrarda optimal sonuç verdiđi kombinasyonların 20 parçalık örneklerde yer alan parametre kombinasyonlarıyla aynı olmasıdır. Bu aralıklarda üretilen parametrelerde sezgisel daha güçlü çalışmaktadır. GAMS ile elde edilen sonuçların hata oranları artarken sezgisel ile elde edilen sonuçların yüzde uzaklıkları azalmıřtır. Sezgisel algoritma 100 parçanın bulunduđu örneklerde hem 20 parçanın bulunduđu hem de 50 parçanın bulunduđu durumlardan daha iyi sonuç vermiřtir. Parça sayısının artmasıyla sezgiselin yüzde hatasının ters orantılı olması büyük problemlerin çözümü için avantaj sağlar ki aslında istenen de budur. Küçük problemlerde zaten optimal sonuca ulařılabilmektedir. Nitekim bu analizde de aynısı olmuřtur.

Tablo 3.7: 100 Parça İçin Özet Tablo

n	Faktörler				GAMS	Sezgisel	CPU	CPU
	f^1	f^2	f^3	s	Boşluk (%)	Hata (%)	GAMS (sn)	Sezgisel (sn)
100	Y	Y	Y	Y	0.19	0.56	720.28	0.06
	Y	Y	Y	D	0.32	1	720.38	0.06
	Y	Y	D	Y	0	0	0.12	0.06
	Y	Y	D	D	0.01	0.01	180.2	0.05
	Y	D	Y	Y	0	0	0.18	0.05
	Y	D	Y	D	0.03	0.46	180.17	0.04
	Y	D	D	Y	0	0.04	12.77	0.04
	Y	D	D	D	0.13	0.19	360.27	0.03
	D	Y	Y	Y	0	0	0.16	0.04
	D	Y	Y	D	0.26	0.83	540.11	0.04
	D	Y	D	Y	0.04	0.42	540.11	0.03
	D	Y	D	D	0.01	0.64	188.15	0.03
	D	D	Y	Y	0.07	0.75	540.11	0.03
	D	D	Y	D	0.24	2	360.29	0.04
	D	D	D	Y	0.11	0.19	900.1	0.03
	D	D	D	D	0.19	0.7	720.2	0.03
Ortalama					0.1	0.47	372.72	0.04

20 parçanın olduğu durumlarda GAMS ile elde edilen değerlerin hepsi optimal değerlerdir.

Çözüm süreleri incelendiğinde ise beklendiği gibi GAMS ile elde edilen çözüm süreleri hemen hemen her kombinasyonda artmıştır. Sezgiselin çözüm süresinin ise 20 parçalık örneklerin çözüm sürelerinden daha uzun ancak 50 parçalık örneklerin çözüm sürelerinden ise daha kısa olduğu görülmüştür. Böyle bir durumla karşılaşılması da yine sezgiselin çözüm süresinin parça sayısına duyarlı olmadığına işaret etmektedir. Bunu sezgisel yöntemin güçlü bir özelliği olarak görmek mümkündür. Dolayısıyla sezgisel algoritmanın çok daha büyük problem verileri için de oldukça kısa sürede çözüm vermesi beklenmektedir. Sezgiselin 100 parçalık örnekler için hem çözüm süresi hem yüzde uzaklık olarak verdiği değerler 50 parçalık için alınan değerlerden daha iyidir.

Bu sonuçlara dayanarak geliştirilen sezgisel yöntemin çok kısa sürede, optimale oldukça yakın sonuçlar veren bir algoritma olduğu açıktır. Algoritmalarda en büyük problem olan veri boyutunun büyümesi ile çözüm süresinin bir anda artması geliştirilen yöntem için geçerli olmayıp sezgiselin en güçlü yönü olarak sunulabilir. Ancak bu problem için hem kısa sürede hem de kesin çözüm elde edilmesi istendiği için başka yöntem daha geliştirilmiştir. Bir sonraki bölümde bu amaca ulaşmayı sağlayan dinamik programlama algoritması anlatılmaktadır.

3.4 Dinamik Programlama Algoritması

Sezgisel algoritmanın optimale oldukça yakın sonuçları çok kısa sürelerde verebildiği önceki bölümde gösterilmiştir. Bu durum optimal sonuca yine kısa sürede ulaşabilen başka bir algoritma geliştirilebilir mi sorusunu beraberinde getirdiği için bir dinamik programlama algoritması geliştirilmiştir. Bu bölümde, üç makineli esnek akış tipi üretim sisteminin sınırsız ara stok kapasiteli durumu için optimal sonuç veren dinamik programlama algoritmasının işleyişi anlatılmaktadır.

Dinamik Programlamanın İşleyişi

Bu bölüm boyunca Önteorem 3.2'de belirlenen kolay durumlar dışında kalan durumlar ele alınacaktır. İşlem zamanlarının (f^j, s) tamsayı olduğu varsayılmaktadır. Bu tip problemlerde genelde yapıldığı üzere gerçekte tamsayı olmayan değerler uygun katsayılarla çarpılarak tamsayı haline getirildikten sonra geliştirilen çözüm yöntemi kullanılabilir.

Dinamik programlamada kullanılacak notasyon aşağıdaki gibi olacaktır:

S_i : i 'inci aşamadaki durum ($i = 1, \dots, n$). S_i durumu 3 makineli bir sistemde (a_i, b_i) şeklinde ikili değerler olarak ifade edilecektir. a_i değeri, i işinin 2'inci makinedeki tamamlanma zamanından 1'inci makinedeki tamamlanma zamanı çıkarılarak, b_i ise i işinin 3'üncü ve 2'inci makinedeki tamamlanma zamanları arasındaki fark hesaplanarak bulunur.

x_i : i 'inci aşamadaki karar değişkeni. x_i (1,2,3) değerlerini alan bir değişken olacaktır. i işi hangi makineye atandı ise x_i o değeri alır.

$q(S_i, x_i)$: i 'inci aşama S_i durumunda ise ve x_i kararı verilirse S_{i+1} 'in elde edeceği değerdir.

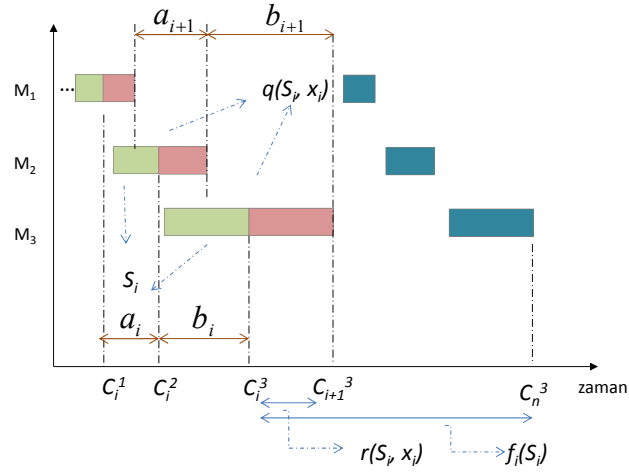
$r(S_i, x_i)$: i 'inci aşamada S_i durumu varken x_i karar değişkenine göre i ile $i + 1$ parçasının makine 3'teki tamamlanma zamanları arasındaki fark $(C_{i+1}^3 - C_i^3)$.

$f_i(S_i)$: i 'inci aşamada S_i durumu varken i parçası ile n parçasının tamamlanma zamanları arasındaki minimum fark $(C_n^3 - C_i^3)$.

Daha açıklayıcı olması için Şekil 3.10'da S_i , $q(S_i, x_i)$, $r(S_i, x_i)$ ve $f_i(S_i)$ değerleri çizelge üzerinde gösterilmiştir.

i parçasının bulunduğu durumda verilen yenileme denklemi ile i parçasından sonraki parçaları içeren minimum tamamlanma zamanı değeri elde edilecektir.

$$f_i(S_i) = \min_{x_i \in \{1,2,3\}} \{r(S_i, x_i) + f_{i+1}(q(S_i, x_i))\} \quad (3.8)$$



Şekil 3.10: Dinamik Programlama Notasyonu

Dinamik programlamanın kullanılabilmesi için bilinmesi gereken başlangıç durumu ve belirlenmesi gereken bir yenileme denklemi vardır. Yenileme denklemine 3.8 numaralı denklemde yer verilmiştir. Dinamik program son parçadan başlayarak ilk parçaya doğru hareket ettiğinden başlangıç durumu Önteorem 3.3'e dayanarak belirlenmektedir. x_n karar değişkeni değerinin 1 olduğu bilinerek işleme başlanır. n 'inci parçanın atamasının makine 1'de olduğu bilindiğinden bu parça için olabilecek bütün $S_n = (a_n, b_n)$ değerleri için bu atama yapılarak $f_n(S_n)$ değerleri bulunur. Her aşamadaki toplam tamamlanma zamanı değerleri ise bir önceki parça için hesaplanan tamamlanma zamanı değerine eklenerek elde edilir. İlk parçaya gelindiğinde bu parçasının atamasının da son makinede olduğu bilinerek hesaplamalar yapılır ve optimal tamamlanma zamanına ulaşılır.

Dinamik programlamanın genel olarak işleyişi şöyle özetlenebilir:

- a_i ve b_i 'nin alabileceği bütün değerler belirlenir.
- Önteorem 3.3'e dayanarak son iş makine 1'e atanır.
- Her parçanın esnek işlem atamasını belirlemek için, son parçadan geriye doğru gidilir. Her durumda o parçadan sonraki minimum toplam tamamlanma zamanları belirlenir. Her aşamada sadece bir önceki aşamada yapılmış hesaplamalar kullanılır. İlk parçaya gelene kadar aynı şekilde devam edilir.
- Önteorem 3.1'e dayanarak ilk iş makine 3'e atanır.

Geliştirilen dinamik programlama yönteminin çözüm süresi, problemde kullanılan parametrelere göre değişmektedir. Çünkü a ve b 'nin alabileceği değerlerin sayısı işlem sürelerine f^j ve s 'e bağlıdır. Bununla birlikte üretilecek parça sayısı da problemin çözüm süresini etkilemektedir.

Dinamik programlamanın temelini oluşturan $S_i = (a_i, b_i)$ durumundaki a_i ve b_i değerlerinin alabilecekleri değerlerin sınırlı olması gerekmektedir. Değilse, geliştirilen algoritmanın çözüm zamanı çok yüksek olacak ve kullanışlı olmayacaktır. a_i değeri en küçük olarak makine 2'nin işlem zamanına eşit olan f^2 değerini alabilir. Bütün parçaların esnek işleminin ikinci makineye atandığı varsayılırsa, Önteorem 3.2'de bahsedilen kolay durumlar haricinde yani $f^2 + s > f^1$ olduğu durumda makine 1 ve makine 2 arasındaki tamamlanma zamanı farkının en büyük değer aldığı durum olacaktır. Dolayısıyla bu değer n . parçada en yüksek değerini alacak ve $n(f^2 + s - f^1) + f^1$ olacaktır. Sonuç olarak a_i 'nin alabileceği değerler $[f^2, n(f^2 + s - f^1) + f^1]$ aralığındaki tüm tamsayı değerler olacaktır. Benzer bir şekilde b_i 'nin değer alacağı aralık için alt sınır f_3 değeri olurken, üst sınır ise esnek işlemlerin hepsinin üçüncü makineye atandığı durum olarak hesaplanacaktır. Buna göre b_i değeri $[f^3, n(f^3 + s - f^2) + f^2]$ aralığında değer alabilecektir.

Önteorem 3.3'e göre başlangıç durumu $x_n = 3$ 'tür. n . parçanın ataması için $f_n(S_n)$ değeri olabilecek bütün $S_n = (a_n, b_n)$ ikililerine $x_n = 3$ uygulanarak bulunur. Böylece i . parçadan n . parçaya kadar olan minimum toplam tamamlanma zamanı, i . parçanın tamamlanma zamanı ile işlem yapılmış bir önceki aşama olan $i + 1$. parçadan n . parçaya kadar elde edilen, minimum toplam tamamlanma zamanların toplanması ile bulunmaktadır. Örnek 2 ile dinamik programlamanın işleyişi daha iyi anlaşılacaktır.

Örnek 2: 4 parçanın üretildiği bir sistemde, sabit işlem süreleri sırasıyla 3,2,4 ve esnek işlem süresi 3 olsun.

Çözüm:

Adım 1: a_i ve b_i 'nin alabileceği bütün değerler belirlenir.

$$a_i \in [f^2, n(f^2 + s - f^1) + f^1] \rightarrow a_i \in [2, 11]$$

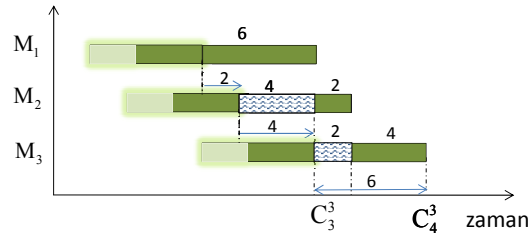
$$b_i \in [f^3, n(f^3 + s - f^2) + f^2] \rightarrow b_i \in [4, 22]$$

Aşağıda verilen tablodan değerlerin nasıl kullanılacağı daha anlaşılır görülmektedir.

Tablo 3.8: Örnek 2 İçin (a_i, b_i) 'nin Alabileceği Değerler

	1	2	3	4
(2,4)	x	-	-	-
(2,5)	x	-	-	-
(2,6)	x	-	-	-
(2,7)	-	-	-	-
...
(2,22)	x	-	-	-
(3,4)	x	-	-	-
...
(9,4)	x	-	-	-
(9,5)	x	-	-	-
...
(11,22)	x	-	-	-

Adım 2: Dinamik programlama son parçadan başlayarak ilerlediği tablo sondan doldurulmaya başlanır. Önteorem 3.3'e dayanarak son işin esnek işlemi ilk makineye atanır. Bu durumda 4. parçanın makinelerdeki işlem süreleri sırasıyla 6, 2 ve 4 olacaktır. İlk durum olan (2,4) ile son parçadan elde edilecek tamamlanma zamanı farkı değeri hesaplanır. Değerlerin nasıl elde edildiği Şekil 3.11'de gösterilmektedir.



Şekil 3.11: Son parçanın (2,4) Durumunda Tamamlanma Zamanı Farkı

Aynı işlem parçanın esnek işlem ataması sabit kalmak üzere tablodaki bütün durumlar için tekrar edilir. Elde edilen tamamlanma zamanı farkı değerleri tabloya not edilir. Tablo 3.9'da 4. parça için elde edilen $f_i(S_i)$ değerlerine yer verilmiştir.

Adım 3: Bir önceki parça olan 3. parçaya geçildiğinde, son parçadan farklı bir durumla karşılaşılır. Bu parçanın atamasının hangi makinede olduğu

Tablo 3.9: Parça 4 İçin Elde Edilen Tamamlanma Zamanı Farkları

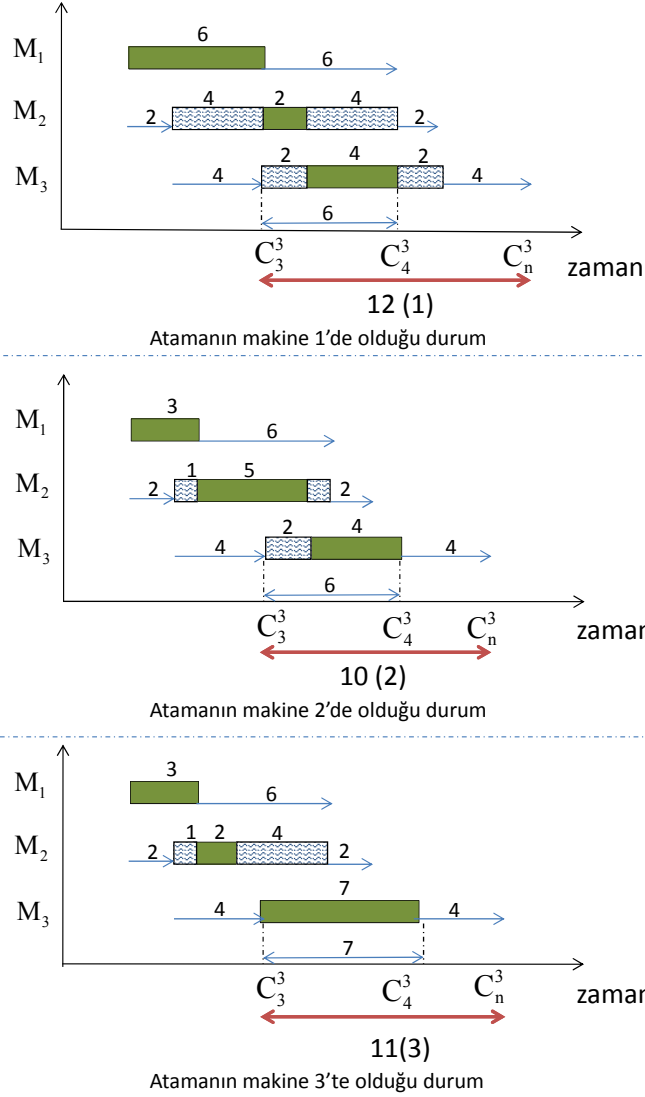
Durum	Parça 4
(2,4)	6
(2,5)	5
(2,6)	4
(2,7)	4
...	...
(2,22)	4
(3,4)	4
...	...
...	...
(9,4)	4
...	...
...	...
(11,12)	4

bilinmediği için bütün durumlar her üç makineye yapılacak atamalarla değerlendirilir. Her durum için en küçük tamamlanma zamanı farkını veren atama değeri ve hangi makine olduğu bilgisi ile birlikte not edilir. Şekil 3.12’de sırasıyla 3. parçanın (2,4) durumunda esnek işlem atamasının makine 1, makine 2 ve makine 3’e yapıldığı durumlarda elde edilen çizelgelere yer verilmiştir. Çizelgelerde yer alan dolgulu alanlar meydana gelen boş zamanları göstermektedir. Aynı zamanda elde edilen $f_i(S_i)$ değerlerinin yanında parantez içinde verilen bilgilerde atamanın hangi makinede olduğu bilgisidir. En küçük tamamlanma zamanı farkı seçilerek tabloda boş alana not edilecektir.

Bu durumda en küçük tamamlanma zamanı farkına sahip olan esnek işlemin makine 3’e olan ataması elde edilen değer ve hangi makine olduğu bilgisiyle birlikte tabloda yerini alır. Tablo 3.10 bütün durumlar için, her üç atama da yapılarak en küçük tamamlanma zamanı farkının değerlerinin seçilmesiyle doldurulur.

Aynı işlemler parça 2 için de tekrar edilerek elde edilen değerler tabloda yerini alır.

Adım 4: İlk parçanın atamasının son makinede olduğu Önteorem 3.1’den bilinmektedir. İlk parçanın tabloda yerleşeceği durum da bilinmektedir ve



Şekil 3.12: 3. Parçanın Atamasının Makine 1, 2 ve 3'te olduğu durum

tek bir alternatif vardır $(a_1, b_1) = (2, 7)$. Parça 2 için elde edilen tamamlanma zamanı değerleri ile ilk parçanın üç makinedeki işlem zamanları $(f^1 + f^2 + f^3 + s = 3 + 2 + 4 + 3 = 12)$ toplanır. Böylece optimal yayılma zamanı elde edilmiş olur. Tablo 3.11'de elde edilen optimal çözümün nasıl bir yol izlediğine yer verilmiştir. Bu durumda ilk parçanın esnek işlem ataması makine 3'e, ikinci parçanın makine 2'ye, üçüncü ve son parça olan dördüncü parçanın ise makine 1'e yapılmıştır. Yayılma zamanı (C_{max}) 24 olarak bulunmuştur.

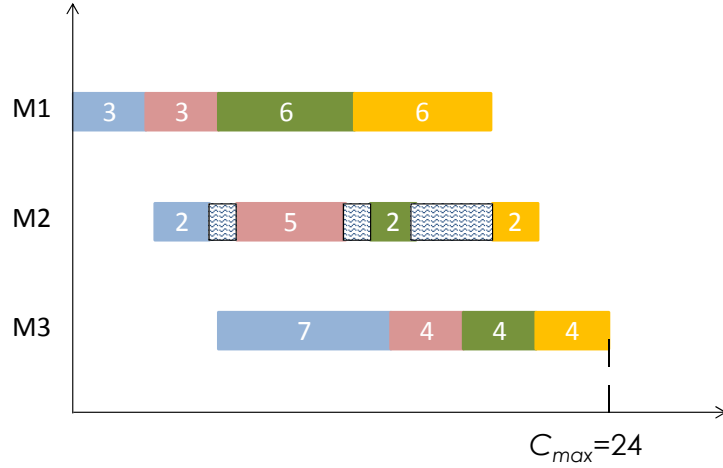
Elde edilen optimal çizelgeye ise Şekil 3.13'te yer verilmiştir.

Tablo 3.10: Parça 3 İçin Elde Edilen Tamamlanma Zamanı Farkları

Durum	Parça 3	Parça 4
(2,4)	10 (3)	6 (1)
(2,5)	-	5 (1)
(2,6)	-	4 (1)
(2,7)	-	4 (1)
...
(2,22)	-	4 (1)
(3,4)	-	4 (1)
...
...
(9,4)	-	4 (1)
...
...
(11,12)	-	4 (1)

Tablo 3.11: Örnek 2 İçin Son Tablo

Durum	Parça 1	Parça 2	Parça 3	Parça 4
(2,4)	X	-	-	-
(2,5)	X	-	-	-
(2,6)	X	-	8(1)	-
(2,7)	24(3)	-	-	-
...	X
(4,2)	X	-	-	4(1)
...	X
(5,5)	X	12(2)
...
(6,4)	X	-	-	...
...
...
(11,12)	X	-	-	-



Şekil 3.13: Örnek 2 İçin Optimal Çizelge

Dinamik programlama algoritması ile kesin çözüm elde edilebilmektedir. Ancak algoritmanın çözüm süresi, problem parametrelerine bağlıdır. En önemli faktör a ve b 'nin değer alabileceği aralıkların boyutudur. Çünkü algoritmanın çözüm süresi a ve b 'nin değer alabileceği aralıkların boyutunun, işlem yapılması gereken $(n - 2)$ adet parça sayısı ve makine sayısı olan 3 ile çarpılmasıyla bulunur. Algoritmanın çözüm süresi şu şekildedir:

$$O((n - 1)^3(f^2 - f^1)(f^3 - f^2) + (n^2s - ns)(n - 1)(f^3 - f^1) + (n^3s^2 - n^2s^2))$$

Örnek 2 gibi çok küçük problem parametrelerinin olduğu problemde dahi optimal sonuca ulaşabilmek için 486 ayrı hesaplama yapılmıştır. Çalışmanın bundan sonraki kısmında a_i ve b_i 'nin alabileceği değer sayısının gerçekte çok daha düşük olduğu gösterilecektir.

Teorem 3.1 *Eğer $b_i > f^3 + 3s$ ise, i parçasının esnek işlemi makine 3'e atanmaz.*

İspat: $b_i > f^3 + 3s$ olduğu durumda, i parçasının esnek işleminin makine 3'e atandığı optimal bir çizelge olduğunu varsayalım. İspat için aşağıdaki aksi durumlar ele alınacaktır.

Durum 1: Eğer $a_i \geq f^2 + s$ ise, i parçasının esnek işlemini makine 3'ten makine 1'e kaydıralım. Yeni çizelgedeki tamamlanma zamanı değerlerini ise \hat{C}_i^j ile gösterelim.

h_1 , i parçasından sonra tamamlanma zamanı değeri $\hat{C}_{h1}^2 > C_{h1}^2$ olan ilk parça, h_2 ise i parçasından sonra tamamlanma zamanı değeri $\hat{C}_{h2}^3 > C_{h2}^3 - s$ olan ilk parça olsun. Bunları kullanarak $h = \min\{h_1, h_2\}$ olacak şekilde bir h parçası belirlenir. Eğer bu özellikleri sağlayan bir h parçası yoksa, yeni çizelge optimaldir. Eğer böyle bir h parçası varsa, bu atama değişikliğinden sonra bütün makineler için tamamlanma zamanı değerleri aşağıdaki eşitliklerdeki gibi olacaktır:

$$\begin{aligned} C_j^1 + s &= \hat{C}_j^1 \quad \forall j = i, i+1, \dots, h, & C_h^1 + s &= \hat{C}_h^1 \\ C_j^2 &= \hat{C}_j^2 \quad \forall j = i, i+1, \dots, h, & C_h^2 &\leq \hat{C}_h^2 \\ C_j^3 - s &= \hat{C}_j^3 \quad \forall j = i, i+1, \dots, h, & C_h^3 &\leq \hat{C}_h^3 \end{aligned}$$

Durum 1.1: Eğer i parçasından sonra bütün parçalar makine 3'e atanmış:

Eğer i parçası makine 1'e atanırsa, makine 2 ve makine 3'te bu parça için boş zaman oluşmayacaktır. i parçasından sonra ise:

- Eğer $f^1 \leq f^2$ ise, ikinci makinede boş zaman oluşmaz. a_i değeri her parçada biraz daha artar ancak makine 2'deki tamamlanma zamanında artış olmaz.
- Eğer $f^1 > f^2$ ise, ikinci makinede i ve n parçaları arasında s 'ten küçük bir boş zaman oluşabilir. Bu boş zaman en büyük $f^1 - f^2$ değerini alacaktır ve $f^1 - f^2 < s$ olduğu bilinmektedir. Yani bir parçanın makine 1'deki başlama zamanı ile makine 2'deki tamamlanma zamanı arasındaki işlem süresi en fazla f^1 olabilir. f^1 şu şekilde elde edilir $f^1 - f^2 + f^2$. i ve n parçaları arasındaki bütün parçalar makine 3'e atandığı ve $f^3 + s > f^1$ olduğu bilindiği için makine 3'te boş zaman oluşmaz. Ayrıca makine 3'teki tamamlanma zamanı da artmaz.

Sonuç olarak yeni çizelge optimaldir.

Durum 1.2: Eğer i parçasından sonra, i ve h parçaları arasında ilk makineye atanan en az bir parça varsa k olsun, bu durumda i ve h parçaları arasında makine 1'e atanan ilk parçanın esnek işleme makine 3'e kaydırılır. Böylece h parçasının tamamlanma zamanları ilk durumla aynı kalır.

$$\begin{aligned}
C_j^1 + s &= \hat{C}_j^1 \quad \forall j = i, i + 1, \dots, k, & C_h^1 &= \hat{C}_h^1, & C_j^1 &= \hat{C}_j^1 \quad \forall j = k, \dots, h \\
C_j^2 &= \hat{C}_j^2 \quad \forall j = i, i + 1, \dots, k, & C_h^2 &= \hat{C}_h^2, & C_j^2 &= \hat{C}_j^2 \quad \forall j = k, \dots, h \\
C_j^3 - s &= \hat{C}_j^3 \quad \forall j = i, i + 1, \dots, k, & C_h^3 &= \hat{C}_h^3, & C_j^3 &= \hat{C}_j^3 \quad \forall j = k, \dots, h
\end{aligned}$$

Bu atama değişikliğinden sonra, h parçasının makine 1'deki tamamlanma zamanı s kadar azalır, makine 2'nin tamamlanma zamanı s 'ten daha küçük bir miktarda azalır, makine 3'teki tamamlanma zamanı ise değişmez. Yeni çizelgenin yayılma zamanı orjinal çizelgenin yayılma zamanına eşit olur.

Durum 1.3: Eğer i parçasından sonra, i ve h parçaları arasında en az bir parça makine 2'ye atanmış ve makine 3'e atanan hiç parça yoksa. Bu parçanın ataması makine 2'den makine 3'e değiştirilir. Bu durumda h parçasının tamamlanma zamanı değerleri aşağıdaki gibi olacaktır.

$$\begin{aligned}
C_j^1 + s &= \hat{C}_j^1 \quad \forall j = i, i + 1, \dots, h, & C_h^1 + s &= \hat{C}_h^1 \\
C_j^2 &= \hat{C}_j^2 \quad \forall j = i, i + 1, \dots, h, & C_h^2 &\leq \hat{C}_h^2 \\
C_j^3 - s &= \hat{C}_j^3 \quad \forall j = i, i + 1, \dots, h, & C_h^3 &\leq \hat{C}_h^3
\end{aligned}$$

h parçasının makine 1'deki tamamlanma zamanını azaltmak için, i parçasının ataması makine 2'ye kaydırılır. Tamamlanma zamanları şu şekilde değişir:

$$\begin{aligned}
C_j^1 + s &= \hat{C}_i^1 & \forall j = i, i+1, \dots, h, & & C_h^1 &= \hat{C}_h^1 \\
C_j^2 &= \hat{C}_i^2 & \forall j = i, i+1, \dots, h, & & C_h^2 &= \hat{C}_h^2 \\
C_j^3 - s &= \hat{C}_i^3 & \forall j = i, i+1, \dots, h, & & C_h^3 &= \hat{C}_h^3
\end{aligned}$$

h ve n parçaları arasındaki tamamlanma zamanları, çizelgenin ilk halindeki tamamlanma zamanlarına eşittir. Yani, yeni çizelgenin yayılma zamanı orjinal çizelgenin yayılma zamanına eşittir.

Durum 1.4: Eğer i ve h parçaları arasındaki bütün parçalar makine 3'e atanırsa, Durum 1.1'e dayanarak makine 3'te herhangi bir boş zaman oluşmaz. h parçası, i parçasından sonraki $\hat{C}_j^2 > C_j^2$ şartını sağlayan ilk parça olabilir. k parçası ise h parçasından sonra makine 3'te boş zaman oluşmasına sebep olan ilk parça olsun.

- Eğer $f^1 \leq f^2$ ise, her zaman $C_j^2 < \hat{C}_j^2$ olacaktır, yeni çizelge optimaldir.
- Eğer $f^1 \geq f^2$ ise, h ve k parçaları arasında bir parça makine 1'e atanmışsa, bu parçanın ataması değiştirilerek makine 3'e alınır. h ve k parçaları arasında boş zamanlar oluşabilir. Ama k parçasının makine 1'deki tamamlanma zamanı değişmez. Dolayısıyla k parçasının makine 2'deki tamamlanma zamanı da değişmez.

$$\begin{aligned}
C_i^1 + s &= \hat{C}_i^1, & C_h^1 + s &= \hat{C}_j^1, & C_k^1 + s &= \hat{C}_k^1 \\
C_i^2 &= \hat{C}_i^2, & C_h^2 &\leq \hat{C}_j^2, & C_k^2 + s &\leq \hat{C}_k^2 \\
C_i^3 - s &= \hat{C}_i^3, & C_h^3 &= \hat{C}_j^3, & C_k^3 + s &\leq \hat{C}_k^3
\end{aligned}$$

Durum 2: Eğer $a_i < f^2 + s$ ise, i parçası makine 2'ye atanır.

Bu durumda makine 1'de değişiklik olmaz, makine 2'deki tamamlanma zamanları s kadar artabilir.

Durum 2.1: i parçasından sonraki bütün parçalar makine 3'e atanmışsa yeni çizelge optimaldir. $f^3 + s > f^2$ olduğu bilindiğinden bu değişiklik, makine 3'te asla boş zaman oluşturmayacak, ve makine 3'teki yeni yayılma zamanı değeri ya aynı kalacak ya da s birim kadar daha küçük olacaktır. Yeni çizelge optimaldir.

Durum 2.2: i parçasından sonra makine 2'ye atanan en az bir parça varsa;

öyle ki böyle bir durumda k parçası makine 3'te boş zaman oluşmasına sebep olan ilk parça olsun.

- Eğer böyle bir k parçası yoksa yeni çizelge optimaldir.
- Eğer böyle bir k parçası varsa i ile k parçaları arasındaki makine 2'ye atanan ilk parçanın esnek işlemi makine 3'e alınır. k parçasının tamamlanma zamanı değerleri aynı kalmış olur. Yeni çizelge optimaldir.

$$\begin{aligned} C_i^1 &= \hat{C}_i^1, & C_h^1 &= \hat{C}_h^1 \\ C_i^2 + s &= \hat{C}_i^2, & C_h^2 &= \hat{C}_h^2 \\ C_i^3 - s &= \hat{C}_i^3, & C_h^3 &= \hat{C}_h^3 \end{aligned}$$

□

Bu teorem sonucunda b değerinin hiçbir zaman $f^3 + 3s$ 'ten daha büyük olamayacağı gösterilmiştir. Dolayısıyla $b_i \in [f^3, f^3 + 3s]$ 'tir ve b 'nin alabileceği değer sayısı $3s$ 'e düşürülmüş olur. Bu ifade halen esnek işlem süresinin büyüklüğüne (s) bağlı olduğu için geliştirilen algoritmanın yapay polinom zamanlı olacağı söylenebilir.

Teorem 3.2 *Eğer $a_i > f^2 + 3s$ ise, i parçasının esnek işlemi makine 2'ye atanmaz.*

İspat: $a_i > f^2 + 3s$ olduğu durumda, i parçasının esnek işleminin makine 2'ye atandığı optimal bir çizelge olduğunu varsayalım.

Durum 1: i parçasından sonra makine 1'e başka herhangi bir parçanın esnek işlemi atanmadıysa:

Bu durumda $f^1 \leq f^2 + s$ ve $f^1 \leq f^3 + s$ olduğu bilindiğinden i parçasının esnek işleminin makine 2'den makine 1'e kaydırılması, makine 3'deki tamamlanma zamanlarını artırmayacaktır. Dolayısıyla yeni çizelge de optimal olacaktır.

Durum 2: i parçasından sonra en az 1 parçanın esnek işlemi makine 1'e atadıysa:

Bu durumda, iki alt başlık altında incelenebilir:

Durum2.1: $f^1 \leq f^2$ ise: Bu durumda, $a_i > f^2 + 3s$ olduğu bilindiğinden, esnek işlem makine 1'e atanmadığı sürece makine 2'de boş zaman oluşmayacaktır. i parçasından sonra esnek işlemin makine 1'e atandığı ilk parça h olsun. O zaman h parçasına kadar yeni C_j^2 değerleri eski C_j^2 değerlerinden s kadar eksik olacaktır. Makine 1'deki değerler s kadar fazla olacaktır. Makine 3'dekiler ise aynı olacaktır. h parçasının ataması makine 1'den makine 2'ye değiştirildiğinde eski haline dönecektir. Dolayısıyla yeni çizelge optimaldir.

Durum 2.2: $f^1 > f^2$ ise: i parçasından sonra esnek işlemi makine 1'e atanan ilk parça h olsun. Bu parçaya kadar makine 2'de boş zaman oluşmadıysa, bu parçanın esnek işlemi makine 1'den makine 2'ye kaydırılır ve eski tamamlanma zamanları elde edilir.

Eğer h parçasından önce makine 2'de boş zaman oluştuysa, bu boş zamanın değeri en fazla $f^1 - f^2 < s$ kadar olacaktır. h parçasına kadar bütün parçaların esnek işlemleri makine 2'ye atandıysa böyle bir boş zaman oluşma ihtimali yoktur. O zaman, i ve h parçaları arasında en az bir parçanın esnek işlemi makine 3'e atanmış olmalıdır. Bu parçanın esnek işlemi makine 2'ye kaydırıldığında, makine 2'de oluşan boş zaman engellenmiş olacaktır. Son olarak çizelgeyi eski haline getirmek için esnek işlemi makine 1'e atanmış olan h parçasının esnek işlemi makine 3'e kaydırıldığında tamamlanma zamanları eski haline gelmiş olacaktır. Dolayısıyla yeni çizelge optimal olacaktır.

Ele alınan durumlar olabilecek bütün alternatifleri kapsamaktadır. Her alternatifte, $a_i > f^2 + 3s$ olduğu durumda i parçasının esnek işleminin makine 2'ye atanmadığı optimal bir çizelgenin bulunduğu ispatlanmıştır. \square

Teorem 3.2 ve 3.1 ile a_i ve b_i 'nin $3s$ kadar değer alabileceği gösterilmiştir. a_i ve b_i için alabileceği bütün değerlerin hesaplanmasına ve bütün değerler için

hesaplama yapılmasına gerek olmadığını bu iki teorem göstermiştir. Dinamik programlamanın teoremler aracılığıyla elde edilen yeni çözüm süresi ilk duruma göre oldukça küçük bir değer olacaktır. a_i ve b_i değerlerinin artık sadece $3s$ değer alabildiği ispatlanınca algoritmanın yeni çözüm süresi de $O(ns^2)$ olarak bulunmuştur. Böylece 3 makine problemi için hem oldukça kısa sürede hem de kesin çözüm veren bir algoritma geliştirilmiştir.

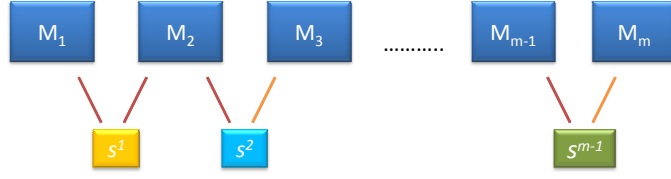
4. m -makine $(m - 1)$ Esnek İşlemlili Akış Atölyesi

Bu bölümde m makine $(m - 1)$ esnek işlemlili probleme ve geliştirilen çözüm yöntemlerine yer verilmektedir. Ayrıntılı olarak problemin özelliklerine değinilerek kurulan karma tamsayıli model anlatılmakta ardından geliştirilirken çözüm yöntemine yer verilmektedir.

4.1 Problem Tanımı ve Matematiksel Model

Bu problem tipinde, n tane özdeş işin üretileceğı varsayılmıştır. Sistemde m adet makine vardır ve parçaların üretiminde, her makinenin sadece kendisinin yaptığı sabit işlemlerin yanı sıra her ardışık iki makine tarafından yapılabilen esnek işlemler de yer almaktadır. Dolayısıyla her parça için m tane sabit işlem ve $(m - 1)$ tane ise esnek işlem yapılmaktadır. Her esnek işlem ardışık iki makine tarafından yapılabilir. İlk esnek işlem makine 1 ve makine 2 yapabilirken, ikinci esnek işlem makine 2 ve makine 3, $(m - 1)$. esnek işlem ise $(m - 1)$ ve m makinesi tarafından yapılabilir. İlk ve son makine hariç diğer makinelerin hepsinde ardışık iki esnek işlem yapılabilir.

Sistemde üretilen parçalar özdeş olduğu için işlem zamanları parçaya göre değışmemektedir. Makinelerdeki sabit işlem zamanları ise birbirlerine göre farklılık gösterebilir. Ancak makineler özdeş olduğu için esnek işlemler hangi makineye atanırsa atansın aynı işlem süresine sahiptir. Makinelerin işlem zamanları f^1, f^2, \dots, f^m , esnek işlemlerin işlem zamanları ise $s^1, s^2, \dots, s^{(m-1)}$ şeklinde tanımlanmıştır.



Şekil 4.1: m -makineli Akış Atölyesi

Bu bölümde de önceki bölümde olduğu gibi ara stok alanının sınırsız olduğu varsayılmıştır. Makinelerde iş kesmeye (preemption) izin verilmez. Makinede belli bir anda sadece bir işin işlenebileceği ve bir işin belli bir anda sadece bir makinede işlenebileceği ise problemin kısıtları arasındadır. Bu problemde verilmesi gereken karar, her bir işe ait bütün esnek işlemlerin hangi makinelere atanacağıdır. Bu atamalar her bir parçadan diğerine farklılık gösterebilmektedir. Problemin amacı son makinedeki son parçanın son makinedeki tamamlanma zamanı olarak tanımlanan yayılma zamanı değerini enküçükmektir.

Bu problemde kullanılacak olan parametreler aşağıdaki gibi olacaktır.

f^j : Makine j 'deki sabit işlem süresi; $j = 1, 2, \dots, m$

s^k : k . esnek işlemin işlem süresi; $k = 1, 2, \dots, m - 1$

Matematiksel modelde yer alan karar değişkenleri ise şu şekildedir.

C_{max} : n . parçanın m makinesindeki tamamlanma zamanı.

T_i^j : i parçasının j makinesinde işleme başlama zamanı.

$$x_{ik}^j = \begin{cases} 1; & i \text{ işinin } k. \text{ esnek işlemi } j \text{ makinesine atanmış ise} \\ 0; & \text{değilse} \end{cases}$$

Bu problem için geliştirilen karma tamsayılı modele aşağıda yer verilmiştir.

$$\min \quad C_{max} \quad (4.1)$$

Öyle ki

$$C_{max} \geq T_n^m + f^m + s^{m-1} \cdot x_{n(m-1)}^m \quad (4.2)$$

$$T_i^1 \geq T_{i-1}^1 + f^1 + s^1 \cdot x_{n(m-1)}^m \quad \forall i \neq 1 \quad (4.3)$$

$$T_i^2 \geq T_i^1 + f^1 + s^1 \cdot x_{i1}^1 \quad \forall i \quad (4.4)$$

$$T_i^j \geq T_{i-1}^j + f^j + s^{j-1} \cdot x_{(i-1)(j-1)}^j + s^j \cdot x_{(i-1)j}^j \quad \forall i \neq 1, \forall j \neq 1, m \quad (4.5)$$

$$T_i^m \geq T_{i-1}^m + f^m + s^{m-1} \cdot x_{(i-1)(m-1)}^m \quad \forall i \neq 1 \quad (4.6)$$

$$T_i^j \geq T_i^{j-1} + f^{j-1} + s^{j-1} \cdot x_{i(j-1)}^{j-1} + s^{j-2} \cdot x_{i(j-2)}^{j-1} \quad \forall i, \forall j \neq 1, 2 \quad (4.7)$$

$$x_{ij}^j + x_{ij}^{j+1} = 1 \quad \forall i, \forall j \neq m \quad (4.8)$$

$$T_1^1 \geq 0 \quad (4.9)$$

$$x_{ik}^j \in \{0, 1\} \quad (4.10)$$

Matematiksel modelde 4.1 amaç fonksiyonunu temsil etmektedir. Amaç fonksiyonu C_{max} değerini enküçüklemeye çalışmaktadır. Kısıtlardan 4.2 numaralı kısıt C_{max} değerinin, son işin son makinedeki başlama zamanına o makinedeki işlem zamanı ve atandıysa esnek işlem zamanının eklenmesiyle oluşan değerden büyük eşit olması gerektiğini belirtmektedir. 4.3 numaralı kısıt i işinin makine 1'de başlayabilmesi için $(i - 1)$. işin makine 1'de sabit işleminin ve atandıysa esnek işleminin tamamlanması gerektiğini belirtmektedir. 4.4 numaralı kısıt i işinin makine 2'de işleme başlayabilmesi için makine 1'de sabit işleminin ve ilk esnek işlem makine 1'e atanmış ise onun tamamlanmış olması gerektiğini belirtir. 4.5 numaralı kısıt ise i işinin j makinesinde işleme başlayabilmesi için $(i - 1)$. işin o makinedeki sabit işleminin ve esnek işlemlerinden atanan olmuşsa onların tamamlanmış olmasını sağlamaktadır. 4.6 numaralı kısıt i işinin m makinesinde işleme başlayabilmesi için $(i - 1)$. işin m makinesindeki işleminin tamamlanmış olmasını sağlamaktadır. 4.7 kısıtı ise makine 3 ile makine m arasındaki herhangi bir makinede bir işin işleme başlamadan önce bir önceki makinede tamamlanmasını sağlamaktadır. 4.8 numaralı kısıt her esnek işlemin atanabildiği ardışık iki makineden birine atanmasını sağlamaktadır. 4.9 numaralı kısıt çizelge başlangıç zamanını belirtmektedir. 4.10 atama karar değişkeninin 0 veya 1 değerini alabileceğini göstermektedir.

Geliştirilen matematiksel model, Intel ® Core™ 2 Quad CPU Q9400 2,66 Ghz, 4 GB Ram, 64 bit bir bilgisayarda, GAMS 23.8.2 arayüzü kullanılarak CPLEX 12.2 çözücüsü kullanılarak çözdürülmüştür. Problem bu matematiksel model ile test edilmiştir. Ancak bu problem, önceki bölümde anlatılan 3 makineli akış atölyesi probleminden çok daha büyük çözüm sürelerine sahiptir. Problemden parça sayısının veya makine sayısının artması çözümü almayı güçleştirmektedir. Özdeş işlerin üretildiğinin varsayıldığı bu tip akış atölyelerinde makine sayısı çok değişken olsa da parça sayılarının çok daha fazla olduğu bilinmektedir. Bu durumlar için matematiksel formülasyonun makul sürelerde çözüm veremediği görülmüştür. Bu sebeple sezgisel bir yöntem geliştirilmeye çalışılmıştır.

4.2 Sezgisel Çözüm Yöntemi

Bu bölümde, m makine ($m - 1$) esnek işlemlili akış tipi atölye probleminin çözümü geliştirilen sezgisel algoritma adımları, algoritmanın test edilmesi için yapılan deneysel çalışmaya yer verilmektedir.

Geliştirilen algoritmada makinelerin iş yükü dengelenmek istenmektedir. Esnek işlemlerin atamaları her makinede birbirine yaklaşık toplam işlem zamanı değerleri elde edilecek şekilde yapılmaya çalışılmıştır. Algoritmada öncelikle her parça için bütün esnek işlemlerin atamalarına karar verilmiştir. Böylelikle her parçanın her makine üzerindeki işlem zamanı belirlenmiştir. Bu belirlenen işlem zamanları ile işlerin farklılaştığı düşünülerek parça sıralaması yapılması için NEH Algoritması [32] yardımıyla işler sıralanmıştır. Algoritmanın çalışma prensibine yer verilmeden hemen önce daha anlaşılır olması için kısaca NEH Algoritması [32]'na değinilecektir:

NEH Algoritmasının başlangıcında işler sıralanır. Her i işinin sistemde bulunan bütün makinelerdeki işlem zamanları toplanır ve işler bu toplama göre artmayan şekilde sıralanır. İlk adımda elde edilen sıralamaya göre ilk iş ilk makinede, ikinci iş ikinci makinede ve tam tersi olacak şekilde işler çizelgelenir. İki ayrı çizelge ve iki ayrı yayılma zamanı değeri elde edilir. Ardından daha küçük yayılma zamanına sahip olan çizelge sabitlenir. Ancak bu çalışmada geliştirilen sezgisel için işlerin sıralanma kuralı bu şekilde olmayacaktır. Önemli olan ve geliştirilen sezgiselde

kullanılan algoritmanın ikinci aşamasıdır. Bu kısımda, işlerin kendi arasında yer değiştirilerek elde edilen çizelgelerden en küçük yayılma zamanına sahip olanın seçilmesi aşamasıdır. Sıradaki işin eklenmesi, k tane olası durum oluşturur ve bu olası durumlar arasından yayılma zamanı en küçük olan sabitlenir. Bu işlem NEH Algoritması'na göre ilk adımda bulunan sıralama göz önünde tutularak 3. sıradaki işten n . sıradaki işe kadar bir sonraki adım tekrarlanır.

Bu çalışmada işlerin sıralanmış şekli farklıdır. Üretilen parçalar özdeş olduğu için makineler üzerindeki işlem süreleri de aynıdır. Parçaları birbirinden farklı kılan esnek işlem atamalarının yapıldığı makinelerdir. Bu nedenle geliştirilen sezgisel yöntemde önce esnek işlem atamalarına belli kurallara göre karar verilmekte ardından NEH Algoritması'nın ikinci kısmı uygulanmaktadır. Algoritmanın ayrıntılı adımlarına Algoritma 2'de yer verilmiştir.

Algoritmanın işleyişi şu şekilde özetlenebilir. İlk ve son iş için esnek işlemler belli kurala göre atanmaktadır. Önteorem 3.1'de verilen optimal çizelge özelliği dikkate alınarak, ilk işin işleminin her makinede bir an önce başlaması esasına dayanan kurala göre her esnek işlem işlenebildiği son makineye atanmaktadır. Böylece makinelerin boş bekleme zamanları en aza indirilmektedir. Son iş için ise tam tersi bir yol izlenerek Önteorem 3.3 dikkate alınarak esnek işlemlerin tamamı atanabildikleri ilk makinelere atanır. İlk ve son iş haricinde ataması belirlenecek ilk parça için, olabilecek bütün olası atamalar ile meydana gelen işlem zamanı değerlerinin ortalaması bulunur. Daha sonra bu ortalama değere yakın işlem zamanı değerleri elde edilmeye çalışılır. Ortalama değere en yakın atama hangi esnek işlem hangi makinede iken elde edilebiliyorsa bu parçanın esnek işlemleri o makinelere atanır. Bu parça da belirlendikten sonra her makine üzerinde mevcut parçaların toplam işlem zamanları toplam iş sayısına bölünerek ortalama değer her parçada güncellenir ve sıradaki parça bu ortalama değere göre atanır. Böylece her makine üzerindeki iş yükü birbirine yakın olmaktadır.

Algoritma 2:

Girdi: f^j, s^j, n, m

Çıktı: C_i^j, C_{max}, x_{ik}^j

1. İlk işin i . esnek işleminin atamasını $i + 1$. makineye yap. $i = 1, \dots, m - 1$

2. Son işin i . esnek işleminin atamasını i . makineye yap. $i = 1, \dots, m - 1$

3. İkinci işin esnek işlem ataması için **Ortalama** hesapla.

4. **Ortalama** = $(2f^1 + s^1 + \frac{\sum_{i=2}^{m-1} (4f^i + 2s^i + 2s^{i-1})}{4m-4} + 2f^m + s^m)$

şeklinde hesapla.

5. **min 1** = $\{|ortalama - f^1|, |ortalama - (f^1 + s^1)|\}$ seç.

6. **min 2** = $\{|ortalama - f^2|, |ortalama - (f^2 + s^1)|, |ortalama - (f^2 + s^1 + s^2)|, |ortalama - (f^2 + s^2)|\}$ seç.

7. **min** değerlerinden elde edilen sonuca göre ilk makinedeki atamalara göre minimum uzaklığı sağlayanı tercih et ve ikinci işin 1.ve m. makinedeki esnek işlem atamasını tamamla.

8. İkinci işin 3. esnek işlemden başlayarak m-1. esnek işlemine kadar atamaları belirlemek için **min 2** değerinde kullanılan formülden yararlanılır.

9. İkinci işin 1.ve m. esnek işlem atamasının belirlenmesi için **min 1** değerinde kullanılan formülden yararlanılır.

10. 1.2.ve n. işlerin esnek işlem atamalarına göre oluşan makineler üzerindeki işlem zamanını yeni bir ortalama değer olarak kullan.

11. Her yeni iş için makineler üzerindeki toplam işlem zamanını yenile ve esnek işlem atması belirlenmiş iş sayısına göre **Ortalama** değeri yenile.

12. Tüm işlerin esnek işlem atamasına yenilenen ortalamaya yakınlık değerine göre karar ver.

13. Belirlenmiş atamalara göre işlem zamanlarını yenile.

14. İşlem zamanları üzerinden tüm işler için NEH algortimasını uygula.

15. İşlerin makineler üzerindeki işlem zamanları toplamı aynı olduğu için rastgele bir sıralama yap.

16. Listenin ilk iki işini önce ilk ardından ikinci iş ve önce ikinci iş sonra ilk iş olacak şekilde 2 türlü çizelgele. Yayılma zamanın düşük olduğu durumdaki sıralamayı sabitle.

17. 3. işten n. işe kadar 18.adım tekrarla.

18. k . iş eklenmesiyle oluşan k tane olası durumdan yayılma zamanı en küçük olanda k sabitlenir.

19. **son.**

4.3 Deneysel Çalışma

Deneysel çalışmada parça sayısının 20 ve 50 makine sayısının ise 5 ve 15 olduğu durumlar incelenmiştir. Parça sayısı da makine sayısı da daha büyük veri boyutlarında incelenmeye çalışılmış ancak hafıza hatası alındığı için matematiksel modelle sonuç elde edilememiştir.

Değerlendirilen bu veriler için ayrı ayrı 10 tekrarlı, sabit ve esnek işlem sürelerinin 3 seviyeye sahip olduğu durumların tüm kombinasyonları ele alınmıştır. Dolayısıyla her problem kümesi için $(3^2 \times 10) = 90$ tane, toplamda 360 problem çözülmüş, sezgiselin performansı bu şekilde test edilmiştir.

Sezgiselin performans analizi sırasında, algoritma ile elde edilen değerler matematiksel model ile elde edilen optimal sonuçlar ile karşılaştırılmıştır. Oluşturulan deney tasarımında işlem süreleri küçük değer aralığı ve büyük değer aralığı olarak belirlenen iki aralıktan türetilerek elde edilmiştir. Kaç makinenin işlem süresi küçük değerlerden kaç makinenin işlem süresi büyük değerlerden türetildi ise deney tasarımı buna göre yapılmıştır. Faktör dizaynında ele alınan seviyeler bu makine sayılarına göre belirlenmiştir. İşlem sürelerinin küçük değerler olduğu aralık $[10, 100]$ iken büyük değerler aralığı $[100, 250]$ olarak belirlenmiştir. İşlem süreleri bu aralıklardan düzgün dağılıma göre türetilmişlerdir. Deney faktörleri dizaynının yer aldığı Tablo 4.1'e aşağıda yer verilmiştir.

Tablo 4.1: Deney Faktörleri Dizaynı

Faktörler		
	f	s
Seviye	Sabit işlem süresi küçük olan makine sayısı	Esnek işlem süresi küçük olan makine sayısı
1	$(0, m/3)$	$(0, m/3)$
2	$(m/3, 2m/3)$	$(m/3, 2m/3)$
3	$(2m/3, m)$	$(2m/3, m)$

Tablo 4.1'de seviyeler sabit işlem süresi ve esnek işlem süresi küçük makine sayılarının türetildiği aralıkları temsil etmektedir. Şöyle ki; f değeri seviye 1, s değeri seviye 2'de olduğu takdirde 5 makineli bir sistemde, sabit işlem süresi küçük aralıktan türetilen makine sayısı $[0, 5/3]$ aralığında olurken, esnek işlem süresi küçük aralıktan türetilen makine sayısı $[5/3, 10/3]$ aralığında olacaktır.

Sabit işlem süresi küçük aralıktan türetilen makine sayısı 0 ile 2 arasında değişecek, esnek işlem süresi küçük aralıktan türetilen makine sayısı 2 ile 4 arasında olacaktır. Böylece sistemde kaç makine varsa geriye kalan makinelerin işlem süreleri de büyük aralıktan türetilmektedir.

İlk test örneği ve en küçük veri seti olan 5 makineli 20 parçanın üretildiği problem verilerine, kaç makinenin sabit ve esnek işlem sürelerinin küçük aralıktan türetildiği, matematiksel modelle belirlenen zaman limitinde optimal sonuca ulaşıp ulaşılamadığı, sezgiselin optimalden (alt sınırdan) % uzaklık değeri ve çözüm sürelerine Tablo 4.2’de yer verilmiştir. İlk üç örnek seti için GAMS çözümü için 3600 saniye zaman limiti konmuştur. Bu limit sadece 15 makine 50 iş probleminde farklıdır.

Tablo 4.2: 5 Makine 20 İş İçin Özet Tablo

m	n	Faktörler	GAMS Boş- luk (%)	Sezgisel Hata (%)	CPU (sn)	GAMS CPU (sn)	Sezgisel CPU (sn)
		(f, s)					
		(1,1)	0	8	15.15		0.01
		(1,2)	0	7	71.46		0.01
		(1,3)	0	4	3.81		0.01
		(2,1)	0	9	93.25		0.01
5	20	(2,2)	0	5	0.88		0.01
		(2,3)	0	2	273.69		0.01
		(3,1)	0	9	256.94		0.01
		(3,2)	0	6	49.27		0.01
		(3,3)	0	4	36.33		0.01
		Ortalama	0	6	88.98		0.01

Tablo 4.2’de yer aldığı üzere 5 makine 20 iş için sezgiselin 90 problem için ortalama uzaklığı %6’dır. Bu örnek seti için zaman limitine takılan problem olmadığından GAMS ile elde edilen bu yüzde uzaklık değeri direkt optimalden olan uzaklığı göstermektedir. Bu tabloda s değerinin 3 olduğu yani esnek işlem süresi küçük aralıktan türetilen makine sayısının 3 olduğu, durumların esnek işlem süresi küçük aralıktan türetilen makine sayısının 1 olduğu durumlardan daha iyi sonuçlar verdiği de görülmektedir. Bu durumda esnek işlem süresi küçük aralıktan türetilen makine sayısı arttıkça sezgiselin daha iyi sonuç verdiğini söylemek mümkündür. Benzer analizi f değerleri için yaptığımızda, f ’nin 1 ve 3 olduğu durumlarda yani

f değerlerinin küçük aralıklardan türetildiği makine sayısı 1 ya da 3 olduğunda, elde edilen % uzaklık değerleri ortalaması aynı iken sabit işlemi küçük aralıktan türetilen makine sayısı 2 olduğu durumun % uzaklık değeri bunlardan daha küçüktür.

Bu örnek seti için sezgisel yöntemin sabit işlem süresi küçük ya da büyük aralıktan türetilen makine sayısına bağlı olmadığı söylenebilir. Ancak bu çok erken bir yorum olur. Diğer örnekler için oluşturulan tablolar da incelenmelidir. Bu problem için çözüm süresinin çok zaman aldığı durumlar çok fazla olmamakla beraber sezgisele göre performansı zaman açısından düşük olarak değerlendirilebilir. İş sayısı arttıkça durumun nasıl değişeceğini gözlemek amacıyla aynı faktörler ve seviyeler 5 makine 50 iş için de kullanılmıştır.

Tablo 4.3: 5 Makine 50 Parça İçin Özet Tablo

m	n	Faktörler	GAMS Boş- luk (%)	Sezgisel Hata (%)	CPU (sn)	GAMS CPU	Sezgisel CPU (sn)
		(f, s)					
		(1,1)	1	9	211.24		0.04
		(1,2)	1	6	2521.92		0.04
		(1,3)	1	5	2160.07		0.03
		(2,1)	1	7	2161.19		0.04
5	50	(2,2)	1	7	442.69		0.03
		(2,3)	1	2	360.11		0.03
		(3,1)	1	9	3240.05		0.03
		(3,2)	1	9	2520.11		0.04
		(3,3)	1	5	1440.34		0.03
		Ortalama	1	7	1673.08		0.03

Tablo 4.2 ve 4.3'ten gözlemlenebileceği üzere 5 makinede işlenecek parça sayısı 20'den 50'ye çıktığında sezgisel hata oranı çok fazla değişmemesine ve çözüm süresi 1 saniye dahi olmamasına rağmen matematiksel modelin ortalama çözüm süresi 90 saniyeden 25 dakikaya çıkmıştır.

Makine sayısını 5 makineden 15 makineye çıkardığımızda ise sezgiselin hata oranı ortalama %6'dan %11'e çıkmıştır. Bu tabloda elde edilen %11 değeri, o problem için optimal sonuç elde edilememiş ise, GAMS ile bulunan alt sınırlardan olan yüzde uzaklığı ifade etmektedir. Bununla birlikte GAMS ortalama çözüm süresi 15 dakikaya yükselmektedir. Aynı zamanda bazı problemlerde çözüm süresi 1 saat

Tablo 4.4: 15 Makine 20 Parça İçin Özet Tablo

m	n	Faktörler	GAMS Boş- luk (%)	Sezgisel Hata (%)	CPU (sn)	GAMS	CPU (sn)	Sezgisel
		(f, s)						
		(1,1)	1	12	1923.49		0.08	
		(1,2)	3	12	405.77		0.01	
		(1,3)	0	9	23.93		0.01	
		(2,1)	1	13	1230.55		0.01	
15	20	(2,2)	0	12	26.81		0.01	
		(2,3)	0	7	1102		0.01	
		(3,1)	2	18	3011.56		0.01	
		(3,2)	1	13	1825.84		0.01	
		(3,3)	0	8	29.64		0.01	
Ortalama			1	11	1064.40		0.02	

sürmüştür. 3600 saniye gibi bir zaman içerisinde çözüme ulaşılamayan problemler olmuştur.

Tablo 4.5: 15 Makine 50 Parça İçin Özet Tablo

m	n	Faktörler	GAMS Boş- luk (%)	Sezgisel Hata (%)	CPU (sn)	GAMS	CPU (sn)	Sezgisel
		(f, s)						
		(1,1)	2	14	900.06		0.11	
		(1,2)	1	12	900.05		0.10	
		(1,3)	1	10	720.09		0.10	
		(2,1)	1	16	900.08		0.10	
15	50	(2,2)	1	15	900.05		0.09	
		(2,3)	1	7	390.66		0.09	
		(3,1)	1	21	810.09		0.10	
		(3,2)	1	17	556.92		0.09	
		(3,3)	1	12	630.16		0.10	
Ortalama			1	14	745.35		0.09	

15 makine için iş sayısı 20 işten 50 işe çıkartıldığında elde edilen sonuçlar 4.5'te verilmiştir. 15 makine 50 iş problemi için alınan koşuturumlar sırasında hafıza problemleri yaşanması nedeniyle, zaman limiti 900 saniyeye düşürülmüştür. Sezgiselimizin bu problem için hata oranı ortalama %14'tür. İş sayısı arttıkça problem çözümünün zorlaşması beklenmiştir. Bu problem için verilen genel

tabloda da belirtildiği gibi zaman limitine takılan problem sayısı oldukça artmıştır.

Tablo 4.6: Bütün Problemleri İçeren Genel Tablo

		GAMS Çözümleri				Sezgisel	CPU	CPU
						Hata (%)	GAMS	Sezgisel
							(sn)	(sn)
m	n	Optimal çözü-	Zaman limitine	GAMS	Boşluk			
		len problem	takılan	(%)				
		sayısı	problem sayısı					
5	20	90	0	0	6	88.98	0.01	
5	50	41	49	0.4	7	1673.08	0.03	
15	20	69	21	1	11	1064.40	0.02	
15	50	15	75	0.7	14	745.35	0.09	
Ortalama				0.5	9	892.95	0.04	

Tablo 4.6'de yer alan genel tablo gözlemlendiğinde zaman limitine takılan problem sayısının, makine sayısından ziyade iş sayısından daha çok etkilendiği görülmektedir. Optimal çözülen problem sayısı 5 makine probleminde, iş sayısı 20'den 50'ye çıkartıldığında 90 problemde 41'e düşerken, iş sayısı 20 kalmak üzere, makine sayısı 5'ten 15'e çıkarıldığında, 90'dan 69'a düşmektedir. 15 makine 50 iş problemi için optimal çözülebilen problem sayısı 15'tir. GAMS çözüm süresi de aynı şekilde hızla artmaktadır. 15 makine 50 iş probleminde çözüm süresinin az görünmesinin sebebi, diğer problemlerde zaman limitinin 3600 saniye, bu problemde ise 900 saniye olmasındandır. Farklı iki zaman limiti kullanılmasının nedeni veri boyutu büyüdüğünde 3600 saniye limitinde matematiksel model yardımıyla sonuç alınamamasıdır. Sezgisel algoritmanın çözüm süresi ise en uzun sürdüğü problemde bile 0.1 saniyedir. Sezgisel yöntemin çözüm süresini ne makine ne de parça artışı önemli derecede etkilememiştir.

Analizin geneline bakıldığında sezgiselin bütün problemler için olan genel performansının optimal ya da alt sınır değerinden yaklaşık %9 uzaklıkta olduğu bulunmuştur. Öte yandan Tablo 4.2 için yapılan s değerlerinin küçük aralıktan türetildiği makine sayısının fazla olduğu durumda sezgisel yöntemin optimal sonuca daha yakın sonuçlar verdiği gözlemi diğer örnek setleri için de doğrulanabilmektedir. Geriye kalan 3 tabloda da görülmektedir ki esnek işlem süresinin küçük aralıktan türetildiği makine sayısı arttıkça sezgisel algoritma daha iyi

sonuç vermektedir. Bunu ilk bakışta veri boyutunun daha küçük olmasıyla ilişkilendirmek mümkün olsa da aynı durum sabit işlem süresinin küçük aralıktan türetildiği makine sayısı için geçerli olmadığı görülmektedir. Bu durum sezgisel algoritmanın esnek işlem sürelerine sabit işlem sürelerine olduğundan daha duyarlı olduğunu göstermektedir.

m makineli sistemin gerçek hayat problemi olarak karşımıza çıktığı düşünüldüğünde matematiksel modelin çok uzun sürelerde sonuç vereceği görülmektedir. Seri üretim yaptığı düşünülen bir sistemde o kadar süre beklemeye sebep olacak bir çözüm yönteminin kullanılması beklenemez.

5. Sonuç ve Değerlendirme

Bu çalışmada esnek işlemin yer aldığı özdeş parça üreten iki ayrı problem ele alınmıştır. Bölüm 3 ve Bölüm 4'te iki ayrı başlık altında incelenen problemlerin önce detaylı bir şekilde problem tanımı yapılmış sonra geliştirilen karma tamsayılı modellerine ve önerilen çözüm yöntemlerine yer verilmiştir.

Bölüm 3'te ele alınan ilk problemde 3 makineli bir akış tipi sistemde parçalar 3 sabit işlemde ve esnek işlemde geçerek ürün haline gelmektedir. Her makinenin kendi sabit işlemi vardır. Aynı zamanda makinelerin hepsi esnek işlemi yapabilme kabiliyetine sahiptir. Problemdeki amaç ise son parçanın makine 3'te tamamlanma zamanını enküçükmektir. Problem için önce karma tamsayılı matematiksel model geliştirilmiştir. Ancak birçok makine çizelgeleme probleminde olduğu gibi veri boyutu büyüyünce matematiksel model ile sonuç almak oldukça zor hale gelmiştir. Bazı örnek problemlerde 12 saati geçen sürelerde sonuç alınmadığı olmuştur. Bu nedenle daha kısa sürede etkin sonuçlar elde edebilmek için dinamik programlama algoritması ve bir de sezgisel bir yöntem geliştirilmiştir. Deneysel çalışmada 240 problem çözülerek yapılan deney tasarımı yardımıyla sezgisel yöntemin performansı test edilmiş ve sonuçlar değerlendirilmiştir.

Bölüm 4'te ise m makine ($m - 1$) esnek işlemin olduğu bir akış tipi sistem ele alınmıştır. Bu problem için de benzer şekilde önce karma tamsayılı model geliştirilmiş, sonuç almak küçük problemlerde dahi çok güç olduğu için ardından sezgisel bir algoritma geliştirilmiştir. Sezgisel yöntemin performansı 320 örnek problem çözülerek test edilmiş ve sonuçlar değerlendirilmiştir.

Her iki problem için de matematiksel model GAMS 23.8.2 arayüzü kullanılarak CPLEX 12.2 çözücüsü ile çözdürülmüştür. Sezgisel yöntemler ise C++ ile

kodlanmıştır.

Sonuç olarak problemler için optimale yakın sonuç veren sezgisel algoritmalar geliştirilmiştir. Ayrıca bu çalışmanın en önemli noktası, 3 makine probleminde optimal sonuca oldukça kısa sürede ulaşan bir dinamik programlama algoritması geliştirilmiş olmasıdır. Her iki problem için de GAMS ile elde edilen matematiksel modelin çözüm kalitesinin de aslında çok kötü olmadığı görülmektedir. Bu noktada akıllara belki, çözümü matematiksel modelle elde etmek gibi bir alternatif olduğu gelebilir. Ancak matematiksel modelin mevcut sonuçlara 3 makine problemi için 15 dakikada, m -makine problemi için ise 1 saatte ulaştığı gözardı edilmemesi gereken bir durumdur. Zira gerçek hayat problemleri düşünüldüğünde böyle bir problemin günde en az bir kere hatta üretiminde olabilecek bazı değişiklikler nedeniyle gün içinde birkaç kez çözülmesi gerekebilecektir. Böyle bir durumda hiçbir firmanın, üretimini 1 saat veya 15 dakika durdurmak gibi bir lüksü yoktur. Zamanla yarışılan bir dönemde, saniyelerin ciddi önemi olduğu unutulmamalıdır. Bu nedenle gerçek hayat problemleri için hem geliştirilen sezgisel algoritmalar hem de dinamik programlama algoritması gerek çözüm kalitesi gerekse çözüm süresi açısından oldukça tatmin edici değerlere sahiptir. Özellikle büyük problem verilerinin olduğu durumlarda geliştirilen sezgisel yöntemleri kullanmak zaman tasarrufu açısından çok büyük katkı sağlayacaktır. Bu çalışmalar ile literatürde ele alınmamış bir problem tipi ele alınarak bir boşluk doldurulmuş aynı zamanda gerçek hayat problemlerine etkin sonuç verebilecek algoritmalar yardımıyla de ışık tutulmuştur.

Gelecekte yapılabilecek çalışmalar olarak ilk akla gelen, problem özellikleri aynı kalmak üzere sıfır ara stok veya sınırlı ara stok kapasitesi durumları olacaktır. Bu durumda her iki problem için de yazılan matematiksel modellere bir kısıt daha eklenecektir. Bu durum haliyle problemin zorluk derecesini artıracaktır. Geliştirilen sezgisel yöntemler üzerinde de değişiklik yapılması gerekecektir. Ya da farklı yöntemler geliştirilerek sonuca ulaşılmaya çalışılabilir. Bu çalışmada m -makine problemi için sadece sezgisel bir yöntem geliştirilmiştir. Belki dinamik programlama algoritması geliştirilerek optimal sonuç elde edilebilir. Ancak bu algoritmanın çözüm süresi çok uzun olacaktır bu nedenle daha kısa sürede çözüm vermesini sağlayan değişiklikler yapılabilir veya mevcut sezgisel yöntem geliştirilerek daha iyi sonuçlar elde edilmeye çalışılabilir.

m -makineli akış atölyesi problemi için ele alınabilecek farklı problem özellikleri de olabilir. Bunlardan birisi ikiden fazla ardışık esnek makine kümelerinin bulunduğu bir problem tipi olabileceği gibi birden fazla esnek işlemin olduğu ve bu esnek işlemlerin sistemdeki belli makinelerin sadece biri ya da daha fazlası tarafından yapılabileceği durumlarda, makinelerin birbirini izleyen bir sırayla konumlandırıldığı problem olabilir. Ardışık iki makinenin esnek olduğu durum bu çalışmada ele alınmıştır. Ele alınabilecek bu problemlerde, bu çalışmada ele alınandan daha farklı bir yol izlemek gerekecektir. Mevcut sezgisel yöntem üzerinde değişikliğe gidilebileceği gibi farklı bir yöntem de geliştirilebilir.

Kaynakça

- [1] Akturk, M.S., Gultekin, H., Karasan,O.E., Robotic cell scheduling with operational flexibility. *Discrete Applied Mathematics*, 145(3):334–348, 2005.
- [2] Allahverdi, A., Gupta J.N.D., Tariq A., A Review of Scheduling Research Involving Setup Considerations, *The International Journal of Management Science*, 27:219-239, 1999.
- [3] Anuar, R., Bukchin, Y., Design and Operation of Dynamic Assembly Lines Using Work Sharing, *International Journal of Production Research*, 44:4043-4065, 2006.
- [4] Askin, R., Chen, J., Dynamic Task Assingment for Throughtput Maximization with Worksharing, *European Journal of Operational Research*, 168:853-869, 2006.
- [5] Babayan A., He D., Solving the n job 3-stage Flexible Scheduling Problem Using an Agent-Based Approach, *International Journal of Production Research*, 42(4):777-799, 2004.
- [6] Baker, K. R., Introduction to Sequencing and Scheduling, *John Wiley and Sons*, 1974.
- [7] Becker C., Scholl A., A Survey on Problems and Methods in Generalized Assembly Line Balancing, *The Journal of Industrial Engineering*, 168:694-715, 2006.
- [8] Boysen N., Fliender, M., Scholl A., A Classification of Assembly Line Balancing Problems, *Europen Journal of Operational Research*, 183:674-693, 2006.

- [9] Browne J., Dubois D., Rathmill K., Sethi, S., Stecke K., Types of Flexibilities and Classification of Flexible Manufacturing Systems, *Division of Research Graduate School of Business Administration The University of Michigan*, 367, 1984.
- [10] Campbell, H.G., Dudek, R.A., Smith, M.L., A Heuristic Algorithm for the n Job m Machine Sequencing Problem, *Management Science*, 16(10), 1970
- [11] Cheng, T.C.E., Gupta, N.D.J., Wang G., A Review of Flowshop Scheduling Research with Setup Times, *Production and Operations Management*, 9(3), 2000.
- [12] Crama, Y., Gultekin, H., Throughput Optimization in Two Machine Flowshops with Flexible Operations, *Journal of Scheduling*, 13(3):227-243, 2010.
- [13] Daniels, R., Mazzola, J., Flow Shop Scheduling with Resource Flexibility, *Operations Reserach*, 42(3):504-522, 1994.
- [14] Daniels, R., Mazzola, J., Dailun, S., Flow Shop Scheduling with Partial Resource Flexibility, *Management Science*, 50(5):658-669, 2004.
- [15] Gangadharan, R., Rajendran, C., A Simulated Annealing Heuristic for Scheduling in a Flowshop with Bicriteria, *Journal of Computers and Industrial Engineering*, 27:473-476, 1994.
- [16] Garey, M., Johnson, D., Computers and Intractability: A guide to the Theory of NP-Completeness, *W. H. Freeman*, 1997.
- [17] Gultekin, H., Aktürk, M.S., Karasan, O.E., Cyclic Scheduling of a robotic cell with tooling constraints. *European Journal Operational Research*, 174:777-796, 2006.
- [18] Gultekin, H., Aktürk, M.S., Karasan, O.E., Bi-criteria Robotic Cell Scheduling, *Journal of Scheduling*, 11(6):457-473, 2008.
- [19] Gultekin, H., Scheduling in Flowshops with Flexible Operations: Throughput Optimization and Benefits of Flexibility, *International Journal of Production Economics*, 140:900-9111, 2012.

- [20] Gupta, N.D.J., Koulamas, C.P. ve Kyparisis, G.J., Potts, C.N., Strusevich, V.A., Scheduling Three-Operation Jobs in a Two Machine Flow Shop to Minimize Makespan, *Annals of Operations Research*, 129:171-185, 2004.
- [21] Gupta, N.D.J., Stafford, E., Flow Shop Scheduling Research After Five Decades, *European Journal of Operational Research*, 169:699-711, 2006.
- [22] Gupta, N.D.J., A Functional Heuristic Algorithm for the Flowshop Scheduling Problem, *Operational Research Quarterly*, 22(1):39-47, 1971.
- [23] Hall, N.G., Sriskandarajah C., A Survey of Machine Scheduling Problems with Bloking and No-wait in Process, *Operational Research*, 44:510-525, 1996.
- [24] Hejazi, S.R., Saghafian, S., Flowshop-scheduling Problems with Makespan Criterion: a Review, *International Journal of Production Research*, 43(14):2895-2929, 2005.
- [25] Ho, J.C., Chang, Y.L., A New Heuristic for The n -job, m -machine Flowshop Problem, *European Journal of Operational Research*, 52:194-202, 1991.
- [26] Johnson, S.M., Optimal Two and Three-Stage Production Schedules with Setup Times Included, *Naval Research Logistics Quarterly*, 1:61-68, 1954.
- [27] Jordan, W., Graves, C., Principles on the Benefits of Manufacturing Process Flexibility, *Management Science*, 41(4):577-594, 1995.
- [28] Kimemia, J., Gershwin, S.B., An Algorithm for the Computer Control of Production in Flexible Manufacturing Systems, *IIE Transactions*, 15:353-362.
- [29] Klein, D., Hannan E., An Algorithm for the Multiple Objective Integer Linear Programming Problem *European Journal of Operational Research*, 9(19):378-385, 1982.
- [30] McClain, J.O., Thomas, J., Sox, C., 'On-the-fly' line balancing with very little WIP. *International Journal of Production Economics*, 27:283-289, 1992.
- [31] Muth, E., Reversibility Property of Production Lines, *Management Science*, 25:152-158, 1979.

- [32] Nawaz, M., Enscore, E., Ham, I., A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem, *Omega*, 11:91-95, 1983.
- [33] Nomden, G., van der Zee D.J., Virtual Cellular Manufacturing: Configuring Routing Flexibility. *International Journal of Production Economics*, 112(1):439-451, 2008.
- [34] Ogbu, F.A., Smith, D.K., The Application of Simulated Annealing Algorithm to the Solution of The $n/m/C_{max}$ Flowshop Problem, *Computers and Operations Research*, 17:243-253, 1990.
- [35] Palmer, D.S., Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time — A Quick Method of Obtaining a Near Optimum, *Operational Research Quarterly*, 16(1):101-107, 1965.
- [36] Pinedo, M., Scheduling Theory, Algorithms and Systems, 3. Baskı, *Springer Science Business Media*, Prentice Hall, 2008.
- [37] Rajendran, C., A Heuristic for Scheduling in Flowshop and Flowline-Based Manufacturing Cell with Multi-criteria, *International Journal of Production Research*, 32(11):1541-2558, 1994.
- [38] Sayın, S., Karabatı, S., A Bicriteria Approach to The Two-machine Flowshop Scheduling Problem, *European Journal of Operational Research*,, 103:129-138, 1999.
- [39] Sethi, A., Sethi, S., Flexibility in Manufacturing: A Survey, *The International Journal of Flexible Manufacturing Systems*, 2:289-328, 1990.
- [40] Vairaktarakis G., Lee C., Analysis of Algorithms for Two-Stage Flowshop with Multi-Processor Task Flexibility, *Naval Research Logistics*, 51:44-59, 2004.
- [41] Yeh, W.C., A New Branch and Bound Approach for the $n/2/\text{flowshop}/\alpha F + \beta C_{max}$ Flowshop Scheduling Problem, *Computers and Operations Research*, 26:1293-1310, 1999.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı : İŞLER, Fatma
Uyruğu : T.C.
Doğum tarihi ve yeri : 29.08.1988 Karaman
Medeni hali : Bekar
Telefon : +90 (505) 534 48 94
e-mail : islerfatma@gmail.com

Eğitim

Derece	Eğitim Birimi	Bölüm	Mezuniyet Tarihi
Y. Lisans	TOBB Ekonomi ve Teknoloji Üniv.	Endüstri Müh.	2013
Lisans	Gazi Üniversitesi	Endüstri Müh.	2011

İş Deneyimi

Yıl	Yer	Görev
2011-2013	TOBB Ekonomi ve Teknoloji Üniv.	Burslu Yüksek Lisans Öğrencisi

Yabancı Dil

İngilizce (Çok iyi)

Yayımlar

“Throughput Optimization in a Flowshop with Three Flexible Machines”, International IIE Conference 2013, İstanbul, 26-28 Haziran, 2013.

“A Dynamic Programming Algorithm for Scheduling Flexible Operations in a Three Machine Flowshop”, EURO/INFORMS International Conference 2013, Roma, 1-4 Temmuz 2013.