

**M-MAKİNELİ ESNEK OPERASYONLU AKIŞ TİPİ SİSTEMLERDE
ÇİZELGELEME**

HATİCE HANDE DEMİRTAŞ

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ARALIK 2011

ANKARA

Fen Bilimleri Enstitü onayı

Prof. Dr. Ünver KAYNAK
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Prof. Dr. Ömer SAATÇİOĞLU
Anabilim Dalı Başkanı

HATİCE HANDE DEMİRTAŞ tarafından hazırlanan M-MAKİNELİ ESNEK OPERASYONLU AKIŞ TİPİ SİSTEMLERDE ÇİZELGELEME adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Hakan GÜLTEKİN
Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Doç. Dr. Bülent TAVLI

Üye : Yrd. Doç. Dr. Hakan GÜLTEKİN

Üye : Yrd. Doç. Dr. Ayşegül ALTIN KAYHAN

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.

Hatice Hande DEMİRTAŞ

Üniversitesi : TOBB Ekonomi ve Teknoloji Üniversitesi
Enstitüsü : Fen Bilimleri
Anabilim Dalı : Endüstri Mühendisliği
Tez Danışmanı : Yrd. Doç. Dr. Hakan GÜLTEKİN
Tez Türü ve Tarihi : Yüksek Lisans – Aralık 2011

Hatice Hande DEMİRTAŞ

M-MAKİNELİ ESNEK OPERASYONLU AKIŞ TİPİ SİSTEMLERDE ÇİZELGELEME

ÖZET

Bu çalışmada m makineli n adet parça üreten bir akış tipi üretim sistemi ele alınmıştır. Bu sistemde her parça makinelerde sırayla işlenmektedir. Her bir makinenin parça üzerinde yapması gereken belirli işlemler vardır. Bunun yanında önceden belirlenmiş ardışık iki makine, sabit işlemlere ek olarak, gereken bir işlemi daha yapabilecek kabiliyettedir. Aynı tip parça üretimi yapılan bu sistemde makineler arası stok sınırsız ya da sıfır kapasiteli olabilir. Amaç, yayılma zamanını enküçükleyecek şekilde, esnek işlemlerin her bir parça için hangi esnek makinede yapılacağına karar vermektir. Problemlerle uygulamada sıkça karşılaşılmaktadır. Esnekliği oldukça yüksek olan farklı uç kapasitelerine sahip CNC makinelerin kullanıldığı esnek üretim sistemleri karşılaşılan alanlardan birisidir. Çapraz eğitilmiş işçilerin bulunduğu montaj hatlarında ve PCB kart üzerine devre elemanlarının yerleştirildiği sistemlerde de aynı durumla karşılaşılmaktadır. Problem çözümü için, öncelikle problemin optimizasyon modeli yazılmış ve optimal çözümün özellikleri belirlenmiştir. Belirlenen bu özellikler ile bazı problemlerin kolayca çözülebileceği veya boyutlarının küçültülebileceği görülmüştür. Ancak bu indirgeme durumunda dahi, optimizasyon modelinin çözüm üretmesi uzun zaman almaktadır. Bu nedenle, optimal çözüm için ispatlanan özelliklerden yararlanılarak bir dinamik programlama algoritması geliştirilmiştir. Yapay polinom zamanlı bu çözüm yönteminin de, problem verilerine bağlı olarak, çözüm süresi oldukça uzun olabilmektedir. Bu nedenle, daha kısa sürede kaliteli çözümler üreten sezgisel bir yöntem geliştirilmiş, yöntemin etkinliği deneysel bir çalışma ile gösterilmiştir. Geliştirilen yöntemler, m makineli ve daha genel esneklik tanımına sahip sistemlerin çözümüne yönelik ilk adımı oluşturmaktadır.

Anahtar Kelimeler: Esnek üretim sistemleri, akış tipi üretim sistemi çizelgeleme.

University : TOBB University of Economics and Technology
Institute : Institute of Natural and Applied Sciences
Science Programme : Industrial Engineering
Supervisor : Assist. Prof. Hakan GÜLTEKİN
Degree Awarded and Date : M.Sc. – DECEMBER 2011

Hatice Hande DEMİRTAŞ

**SCHEDULING IN M-MACHINE FLOWSHOPS WITH FLEXIBLE
OPERATIONS**

ABSTRACT

In this study, we consider an m machine flowshop which produces n identical parts. Each machine has to make some preassigned operations. Additionally, a pair of adjacent machines are capable of performing an additional operation, besides their preassigned operations. There may be an infinitive capacity buffer or no buffer in between machines. The problem is to decide on which of the flexible machines to perform this operation. For each part, the objective is to minimize the makespan. Such problems are encountered. In some industries, such as the metal cutting industry where highly flexible CNC machines are used extensively. Similar problems are also encountered in assembly lines where cross-trained operations are working and assembly systems of printed circuit boards (PCB). In order to solve the problem, the optimization model is developed and the characteristics of the optimal solution are determined. Using these characteristics, some problems can be solved trivially or the problem size can be reduced. However, even the reduced problem requires a long time to solve with the optimization model. Therefore, a dynamic programming algorithm is developed which uses the characteristics of the optimal solution. The pseudo polynomial time dynamic programming algorithm also requires a long solution time for some problems depending on problem parameters. As a consequence a heuristic procedure is developed to produce high quality solutions in reasonable times. The performance of the heuristic is tested through a computational study. The developed procedure constitute the first step for the solution of m -machine flowshops with more general flexibility systems.

Keywords: Flexible manufacturing systems, flowshop scheduling.

TEŐEKKÜR

Yüksek lisans eğitimim süresince bana rehberlik eden, değerli katkıları ve yardımları ile çalışmamı yönlendiren tez danışmanım Yrd. Doç. Dr. Hakan GÜLTEKİN'e, tezimi okuyarak tavsiyelerde bulunan jüri üyeleri Doç Dr. Bülent TAVLI ve Yrd. Doç. Dr. Ayşegül ALTIN KAYHAN'a, gerçek dostlukları ve içten destekleri için ofis arkadaşlarım başta olmak üzere tüm asistan arkadaşlarıma ve maddi manevi her zaman yanımda olan aileme teşekkürü bir borç bilirim.

İÇİNDEKİLER

1 Giriş	1
2 Literatür Araştırması	4
2.1 Akış Tipi Üretim Sistemlerinde Çizelgeleme	4
2.2 Üretimde Esneklik	9
2.3 Esnek Akış Tipi Üretim Sistemlerinde Çizelgeleme	13
2.4 Özet	17
3 Problem Tanımı ve Eniyileme Modeli	18
3.1 Problem Tanımı	18
3.2 Esnekliğin Uygulama Alanları	19
3.3 Esnekliğin Getirisi	21
3.4 Problemin Zorluğu	22
3.5 Eniyileme Modeli	24
4 Temel Sonuçlar	27
5 Dinamik Programlama	39
5.1 Dinamik Programlama Genel Durum	39

5.2	Sınırsız Ara Stok Problemi	43
5.2.1	3 Makineli Sistem, Esnek Makinelerin 1 ve 2 Olduğu Durum ($m = 3, k = 1$)	43
5.2.2	3 Makineli Sistem, Esnek Makinelerin 2 ve 3 Olduğu Durum ($m = 3, k = 2$)	49
5.2.3	4 Makineli Sistem, Esnek Makinelerin 2 ve 3 Olduğu Durum ($m = 4, k = 2$)	50
5.2.4	m Makineli Sistem	52
5.3	Sıfır Ara Stok Problemi	56
5.3.1	3 Makineli Sistem, Esnek Makinelerin 1 ve 2 Olduğu Durum ($m = 3, k = 1$)	56
5.3.2	3 Makineli Sistem, Esnek Makinelerin 2 ve 3 Olduğu Durum ($m = 3, k = 2$)	59
5.3.3	4 Makineli Sistem, Esnek Makinelerin 2 ve 3 Olduğu Durum ($m = 4, k = 2$)	60
5.3.4	m Makineli Sistem	61
6	Sezgisel Çözüm Yöntemi	63
6.1	Algoritma Adımları	63
6.2	DeneySEL Çalışma	66
6.3	Sonuçlar ve Değerlendirme	68
6.3.1	Sınırsız ara stok problemi	68
6.3.2	Sıfır ara stok problemi	70
7	Sonuç ve Değerlendirme	73

ŞEKİLLERİN LİSTESİ

3.1	m makineli akış tipi sistem	19
3.2	Örnek 1 için esnek ve esnek olmayan sistemlerin optimal Gantt şemaları	22
3.3	Örnek 2 için r hesabı ve optimal çözümün Gantt şemaları	23
4.1	Önteorem 2- İlk parçanın ataması	29
4.2	Önteorem 3- Durum 1, sınırsız ara stok	31
4.3	Önteorem 3- Durum 1, sıfır ara stok	32
4.4	Önteorem 3- Durum 2, sınırsız ara stok	33
4.5	Önteorem 3- Durum 2, sıfır ara stok	33
4.6	Önteorem 3- Durum 3 $f^h > f^{k+1}$, sınırsız ara stok	35
4.7	Önteorem 3- Durum 3 $f^h \leq f^{k+1}$, sınırsız ara stok	36
4.8	Önteorem 3- Durum 3, sıfır ara stok	37
5.1	4 makineli sistemde (a,b,c) değerleri	40
5.2	4 makineli sistemde notasyonların gösterimi	41
5.3	Örnek 4 için beşinci parçanın (11,12) ve (11,13) durumlarındaki Gantt Şemaları	47
5.4	Örnek 4 için dördüncü parçanın (11,12) durumundaki Gantt Şemaları	48

5.5	Örnek 4 için optimal Gantt Şeması	48
5.6	Örnek 6 için optimal Gantt Şeması	55
6.1	İki parçadan üçüncü parçaya geçişte olası atama alternatifleri	65
6.2	Üçüncü parçadan dördüncü parçaya geçişte olası atama alternatifleri .	66

ÇİZELGELERİN LİSTESİ

5.1	Dinamik programlama Örnek 4-durumlar ve aşamalar	46
5.2	Dinamik programlama Örnek 4-optimal çizelge	49
6.1	Test parametreleri	67
6.2	Sınırsız ara stok-genel	69
6.3	Sınırsız ara stok-esnek işlem sürelerinin türetildiği aralığa göre	69
6.4	Sınırsız ara stok-sabit işlem sürelerinin türetildiği aralığa göre	70
6.5	Sıfır ara stok-genel	71
6.6	Sıfır ara stok-esnek işlem sürelerinin türetildiği aralığa göre	71
6.7	Sıfır ara stok-sabit işlem sürelerinin türetildiği aralığa göre	71

1. Giriş

Teknolojide meydana gelen gelişmeler, müşteri taleplerindeki artan çeşitlilik, ürün yaşam döngülerinin giderek kısalması ve artan global rekabet neticesinde firmalar yaşamlarını sürdürebilmek için üretim sistemlerini esnekleştirmek ve sahip oldukları kısıtlı kaynakları daha verimli kullanmak zorundadırlar. Planlama ve çizelgeleme problemlerinin ortaya çıkış nedeni budur. Her ne kadar esneklik firmaların değişikliklere kolay adapte olmasını sağlayan önemli bir özellik olsa da, çözülmesi gereken önemli stratejik, taktiksel ve operasyonel problemleri de beraberinde getirmiştir. Stratejik ve taktiksel problemler arasında, ne kadar, hangi alanlarda ya da hangi tip esnekliğin uygulanacağı ve ne kadar esnekleştirme maliyetine katlanılacağı yer almaktadır. Operasyonel problemler ise mevcut esneklikten maksimum faydanın nasıl sağlanabileceğine yönelik planlama, çizelgeleme ve üretim faaliyetlerinin nasıl yapılabileceğini ele almaktadır.

Bu çalışma, esnekliğin ortaya çıkardığı operasyonel problemlerden bir tanesini ele almaktadır. Bu kapsamda, n adet iş ve m adet makineden oluşan bir akış tipi üretim sistemi konu alınmıştır. Bir parça makinelerden sırasıyla geçerken belirli bir işleme tabi tutulur. Bunun haricinde, makineler arasında yapması gereken sabit iş dışında başka bir işlem yapabilme kabiliyetine sahip ardışık iki makine vardır. k ve $k + 1$ indisleri ile gösterilecek bu makineler “esnek makine” olarak adlandırılmışlardır. Bu makinelerin yapabildikleri ek işlem de “esnek işlem” olarak adlandırılacaktır. Problem, statik bir şekilde birbirinin aynı olan parçalar için esnek işlemin hangi makinede yapılacağına karar vermek değil, dinamik olarak her bir parça için bu kararı teker teker vermektir. Bunun sonucunda, her bir parçanın ataması birbirinden farklı olabilir. Farklı atamalar, makinelerdeki işlem zamanlarını değiştirecek, bu da farklı çizelgeler ve tamamlanma zamanları ortaya çıkaracaktır. Bu çalışmaya konu olan problemin amacı, son işin son makinedeki tamamlanma zamanı olarak tanımlanan

yayıma zamanını (makespan) enküçükleyecek şekilde, her bir parça için esnek işlem atamasının bulunmasıdır.

Sistemde yer alan makineler arası stok alanı sınırsız ya da sıfır kapasiteli olabilir. Ara stok kapasitesinin sınırsız olması n parçalı bir sistemde $n - 1$ adet stok alanı olması şeklinde düşünülebilir. Sıfır ara stok kapasiteli problemde ise, makineler arasında parçaların konabileceği herhangi bir alan bulunmamaktadır. Bunun sonucunda, herhangi bir makine işlemini bitirdiğinde bir sonraki makinede bir önceki parça işlemine devam ediyorsa bu makine boşaltılamayacak ve dolayısıyla yeni bir parça alıp işlemeye başlayamayacaktır. Bu durum literatürde “bloke olmak” olarak ifade edilmektedir. Sıfır ara stok problemine özel olan bu durum ile sınırsız ara stok probleminde karşılaşmamaktadır. Diğer taraftan, herhangi bir makine bir parça üzerinde işlemini bitirdiğinde, bir önceki makine sıradaki parçayı işlemeyi bitirmediyse bu makine boş olmasına rağmen işlemeye başlayamaz. Bu durum da “aç kalmak” olarak adlandırılmıştır ve bu durumla hem sıfır hem de sınırsız ara stok problemlerinde karşılaşılabilir. Bu problem için bütün işlemlerin zamanları sabittir ve önceden bilinmektedir. Bir parça bir makinede işleme başladığında o iş bitirilene kadar işlem kesilemez. İşlenecek hammadde ya da parçalar sıfır anında giriş stoğunda hazır bulunmaktadırlar.

Ele alınan problemin pratikte uygulamalarının sınırlı olduğu düşünülebilir. Fakat, özellikle makine sayısının 2, 3 ve 4 olduğu benzer özellikteki sistemlerde bu problemle oldukça sık karşılaşmaktadır. Diğer taraftan makine sayısının çok daha fazla olduğu otomotiv gibi endüstrilerde de benzer özellikler bulunmaktadır. Ancak büyük boyutlu bu problemlerde esneklik ilişkileri genelde çok daha karmaşıktır ve problemin çözümü oldukça zordur. Bu çalışmanın amacı, bu genel problemin çözümü için kullanılabilecek öngörüler ve sonuçlar elde etmektir.

CNC makinelerin kullanıldığı esnek üretim sistemlerinde, çapraz eğitilmiş işçilerin çalıştığı montaj hatlarında ve PCB kart üzerine devre elemanlarının monte edildiği sistemlerde problemle sıkça karşılaşmaktadır. CNC makineler yüksek esnekliğe sahip sistemlerdir. Bu makineler uç haznelerinde belirli sayıda uç bulundurabilmektedirler ve gerekli olan uçlar makinede bulunduğu müddetçe, bu uçlara karşılık gelen işlemleri yapabilmektedirler. Hem uç hazneleri limitli kapasitede olduğu için ve hem de mali kısıtlardan dolayı her makinede her ucun bulunması mümkün olmamaktadır. Bazı makinelerde sadece kendi işlemi için gerekli uçlar bulunurken diğer makinelerde kendi

uçlarının yanında diğer işlemlerden bazılarını işleyebilecek uçlar da yer almaktadır. Bu uçların bulunduğu makineler probleme konu olan esnek makineleri ifade etmektedir. Montaj hatlarında iş yüklerinin dengeli dağıtılması için yapılan çapraz eğitim de esnek bir sistem ifadesidir. İşçilerin birbirleri yerine yapabildikleri işlemler esnek işlemleri oluşturmaktadır. PCB kart üzerine devre elemanlarını yerleştiren robot kollarda da benzer durumlar ortaya çıkmaktadır. Her bir devre elemanını yerleştirmek için robot kola farklı bir tutucu yerleştirmek gerekmektedir. Her bir tutucu robot kol üzerinde belirli bir yer işgal etmektedir ve robot kolun belirli bir sayıda fiziksel kapasitesi vardır. Dolayısıyla tek bir robot kola bütün tutucuları yerleştirmek mümkün olmayacağı için, her bir robot kolda aynı ve farklı tutucular bulunabilir. Aynı tutucuların yaptıkları işlemler esnek işlem olarak adlandırılabilir.

Bu problemin çözümü için öncelikle 0-1 tamsayılı bir matematiksel model geliştirilmiştir. Yapılan ilk deneysel çalışmada parça sayısı ve makine sayısı arttıkça matematiksel modelin makul zamanlarda çözüm üretemediği görülmüştür. Bunun üzerine, öncelikle problem detaylı şekilde analiz edilerek optimal çözümün çeşitli karakteristik özellikleri belirlenmiştir. Bu özellikler kullanılarak optimal çözümü yapay polinom zamanda üreten bir dinamik programlama algoritması geliştirilmiştir. Ancak bu yöntemin de bazı büyük boyutlu problemlerde oldukça uzun zamanda çözüm verdiği görülmüştür. Bu sebeple polinom zamanlı olarak çözüm veren sezgisel bir yöntem geliştirilmiştir. Bu yöntem, optimal çözümü bulması garanti eden bir yöntem değildir. Ancak ilerleyen bölümlerde de görüleceği gibi, sezgisel yöntem çok kısa sürelerde optimale oldukça yakın kaliteli sonuçlar vermektedir.

Tez çalışması 6 bölüm halinde incelenecektir. Bir sonraki bölümde akış tipi üretim sistemlerinde çizelgeleme, üretimde esneklik ve problemle ilgili yapılmış çalışmaların derlendiği literatür taraması yer almaktadır. 3'üncü bölümde detaylı problem tanımı, esnekliğin getirisi ve geliştirilen matematiksel model anlatılmaktadır. 4'üncü bölüm, optimal çözümlerin sahip olduğu çeşitli özelliklerin anlatıldığı bölümdür. Bu özellikler esas alınarak geliştirilen dinamik programlama algoritması 5'inci bölümde anlatılırken, 6'ıncı bölümde dinamik program ve matematiksel modelin özellikle büyük parça ve makine sayısı nedeniyle uzun sürede çözüm verdiği durumlar için geliştirilmiş olan sezgisel çözüm yöntemi yer almaktadır. Sezgisel çözüm yöntemi sonuçlarının analizi için yapılan testler de 6'ıncı bölümde yer almaktadır. 7'inci bölüm ise, çalışmayla ilgili sonuçlar ve yorumların yanında problemle ilgili gelecekte yapılabilecek çalışmalar için ayrılmıştır.

2. Literatür Araştırması

Çizelgeleme konusunda ilk çalışmalar 1950'li yıllarda başlamış ve günümüze kadar bu konuda sayısız çalışma yapılmıştır. Bu çalışmalar, gün geçtikçe çeşitlenmiş ve gerçek hayatta karşılaşılan üretim sistemlerini ele almışlardır. Tek makine çizelgeleme problemi ile başlayan çalışmalar, çok makine çizelgeleme başlığı altında akış tipi, atölye tipi, paralel makineler gibi sistemler ele alınarak daha geniş bir alana yayılmıştır. Çalışmalar, son yıllarda ise modern üretim sistemlerinde ortaya çıkan çizelgeleme problemleri üzerinde yoğunlaşmıştır. Bu sistemler arasında robotlu üretim sistemleri, bilgisayarlı bütünleşik imalat sistemleri ve esnek üretim sistemleri sayılabilir.

Bu çalışmanın konusu olan m makineli esnek operasyonlu akış tipi sistemlerde çizelgeleme probleminin literatür araştırması konuyla ilgili 3 başlık altında toplanmıştır. İlk bölümde klasik akış tipi üretim sistemleri, ikinci bölümde üretimde esneklik, son bölümde ise esnek akış tipi üretim sistemleri konularında yapılmış çalışmalardan bahsedilecektir.

2.1 Akış Tipi Üretim Sistemlerinde Çizelgeleme

Çoklu makine çizelgeleme probleminin bölümlerinden biri olan akış tipi üretim sistemi çizelgeleme literatürü, günümüzden yaklaşık 60 yıl kadar önce Johnson [1]'in yaptığı çalışma ile başlar. İzleyen yıllarda farklı problem türleri ile de çalışılmış ve günümüze kadar 1200'ü aşkın makale yöneylem araştırması literatürüne girmiştir. Oldukça fazla çalışma bulunan çizelgeleme alanındaki problem çeşitleri ve çözüm yaklaşımları hakkında daha geniş kapsamlı bilgi için Pinedo [2]'den yararlanılabilir.

Geleneksel akış tipi üretim sistemi Tanaev vd. [3] tarafından şu şekilde tanımlanmıştır:

"Aynı teknolojik sıra ile m makinede yapılması gereken n iş ve i işinin j makinesindeki işlem zamanı P_i^j ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$) verilir. Amaç, n tane işin m tane makinenin her birinde işleneceği üretim maliyetini enküçükleyen sırayı, yani çizelgeyi bulmaktır."

Pinedo [2] tarafından akış tipi üretim sistemi şu şekilde ifade edilmiştir:

"Seri m makine vardır. Her bir iş m makinenin her birinde işlenmek zorundadır. Tüm işler aynı rotayı izlemek zorundadır, örneğin tüm işler önce birinci makinede, sonra ikinci makinede gibi. Bir işin bir makinedeki işlemi tamamlandıktan sonra sonraki makinenin işlem kuyruğuna girer. Genellikle, tüm kuyruklar ilk giren ilk çıkar (FIFO) disiplinine göre çalışır. Makineler arası ara stok kapasiteleri parça türüne bağlı olarak değişmektedir. PCB kartlar, entegre devre üretimi yapan sistemlerde küçük alanlara bir çok parça stoklanacağı gibi, televizyon gibi büyük parça üretimi olan hatlardaki ara stok alanı istasyon tıkanmasına neden olacak kadar az (bazen sıfır) olmaktadır."

Bahsedildiği gibi 1954 yılında Johnson'ın yaptığı çalışma akış tipi çizelgeleme problemlerine öncülük etmektedir. Johnson algoritması, iki makineli ya da başka bir ifade ile her aşamada sadece bir makine bulunan iki aşamalı sistemler için geliştirilmiş sonsuz ara stok seviyesi için optimal sonuç veren bir çözüm yöntemidir. n tane parça önce ilk, sonra da ikinci makineden geçmek zorundadır. Hazırlık süreleri işlem zamanlarına dahildir. Tüm işlerin tamamlanma zamanını enküçüklemeyi amaçlayan m makineli akış tipi üretim sisteminin çizelgelenmesini tasarlama ve 2 makineli sisteme başlangıç ve bitiş zamanları gecikmelerini dahil ederek mevcut problemi geliştirmeyi içerir. $O(n \log n)$ sürede çalışan bu algoritmanın işleyişini, ele alınan problemle yakından ilgili olduğu için, detaylı bir şekilde incelemek faydalı olacaktır. Çözüm adımları aşağıdaki gibidir:

- Her parça için her bir makinedeki işlem sürelerini listele.
- Tüm periyotlar içindeki en küçük zamanı seç.
- Eğer bu süre ilk makinenin sahip olduğu bir işlem süresi ise bunu sıranın en başına yaz.

- Eđer bu süre ikinci makinenin sahip olduđu bir işleml süresi ise bunu sıranın en sonuna yaz.
- Bu süre hangi parçaya aitse o parçayı listeden çıkar.
- Bütün parçalar sıralanana kadar bu işlemlı tekrarlar.

Üç makine ve farklı tip parça problemi ise NP-Zor'dur [4]. Fakat 3 makineli sistemde, ortadaki makinenin en büyük işleml süresinin ilk veya son makinelerin en küçük işleml süresinden küçük olduđu özel durumlarda 2 makine sistemi için geliştirilen algoritma 3 makine sistemi için de optimal çözümü vermektedir [1].

Johnson'dan sonra artan bir ivme ile hızlanan akış tipi üretim sistemlerinde çizelgeleme konusunda yapılmış yüzlerce çalışma Gupta [5] tarafından incelenmiş ve gelişmeler kronolojik olarak şu şekilde özetlenmiştir:

1955-1964 yılları arasında çalışmalar Johnson'ın 2 makineli akış tipi üretim sistemi çizelgeleme konusundaki yaklaşımlarını takip etmiştir. Bu dönemde matematiksel programlama gibi optimizasyon tekniklerinin uygulandıđı teorik çalışmaların yanısıra bir grup araştırmacı da Monte-Carlo benzetim tekniklerini kullanmıştır.

Sonraki 10 yılda çözüm teknikleri gelişmiş ve son işin tamamlanma zamanından farklı olarak toplam tamamlanma zamanı, toplam gecikme zamanı, toplam akış zamanı gibi farklı amaç fonksiyonu düşünceleri ortaya çıkmıştır. Dal sınır çözüm yaklaşımının temelleri bu dönemde atılmıştır.

1975-1984 arasındaki yıllarda NP-Tam hesaplama kuramının açığa çıkması ile akış tipi sistem çizelgelemesi geliştirmeleri derinden etkilenmiştir. Bir yandan çeşitli akış tipi çizelgeleme problemlerinin karmaşıklığı için çalışmalar yapılırken, bir yandan da bir çok sezgisel yöntem geliştirilmiştir. Bir çok farklı çizelgeleme problemi ortaya çıkarken çeşitli varsayımlar da oluşmaya başlamış ve çeşitli problem tipleri için stokastik işleml zamanları da ilk defa bu dönemde ele alınmıştır.

1985 sonrasındaki 10 yılda her bir seviyede birden çok paralel makinenin olduđu hibrit akış tipi sistemler ve tabu arama, genetik algoritma, tavlama benzetimi gibi metasezgiseller ortaya çıkmıştır. Yapay zeka bazlı bazı karar destek ve uzmanlık algoritmaları da bu problem için kullanılmaya başlanmıştır.

1995'den günümüze akış tipi sistem problemleri, amaç fonksiyonları ve çözüm yaklaşımları çeşitleri artmaya devam etmiştir. Çok-kriterli akış tipi sistem çizelgelemesi giderek popülerleşmiştir. Robotik akış tipi sistemler ve AGV'li akış tipi sistemler için çözüm yöntemleri geliştirilmiştir. Bilgisayar verimliliğindeki artış nedeniyle akış tipi çizelgeleme problemleri için matematiksel programlamanın kullanılabilirliği tekrar gündeme gelmiştir.

Özet olarak, literatürdeki çalışmalar genellikle Johnson'ın yaptığı çalışmayı $m > 2$ makineler için geliştirmek üzere yapılmıştır. Dinamik programlama, dal-sınır yöntemi, eleme yaklaşımları, satır indirgeme yöntemleri ve tamsayı programlama gibi çözüm yöntemleri büyük boyutlu akış tipi çizelgeleme problemleri için kullanılamamaktadır. Bu tip problemlerin çözümü için geliştirilmiş bir çok sezgisel yöntem vardır. Kullanılan sezgisel çözüm yaklaşımları, metasezgisellerin mevcut problem için uyarlaması olabileceği gibi; çeşitli şekillerde kurulacak yeni inşa algoritmaları da olabilir. Bu çözüm yöntemlerinin en bilinen ve en yaygın kullanılanları aşağıdaki gibidir [6]:

- **Palmer Algoritması (1965):** Her bir iş için, işlem süresinin küçük işten büyük işe doğru sıralanıp sıralanmadığını ölçen eğim dizisi kavramını kullanır.
- **Gupta Algoritması (1971):** Palmer'in algoritmasına benzer bir yaklaşım sergilemiş, ancak eğim dizisinde farklı bir hesaplama yöntemi geliştirerek Palmer'den daha iyi bir sonuca ulaşmıştır.
- **CDS Algoritması (1970):** Bu algoritma, problemi $m - 1$ adet iki makineli probleme dönüştürür. Johnson Algoritması'nı kullanarak her bir problem için ayrı ayrı çizelge oluşturur ve en iyisini seçer.
- **NEH Algoritması (1983):** Nawaz vd. [4], n iş ve m makineden oluşan akış tipi sistemlerde $(n!)^m$ adet, her bir parçanın makinelerde işlenme sırası değişebilen alternatif çizelgelerin olduğunu, permutasyon akış tipi problemler için ise $(n!)$ alternatif çözüm olduğunu göstermiştir. Bunların yanısıra permutasyon akış tipi çizelgeleme probleminin $m > 2$ için NP-Zor olduğu gösterilmiştir [4]. Bu algoritmada toplamda $n(n + 1)/2 - 1$ adet deneme yapılmaktadır ve adımları aşağıdaki şekildedir:
 - **Adım 1:** Her bir işin tüm makinelerdeki işlem zamanları toplanır. İşler bu toplamlara göre, azalan sırada olacak şekilde listelenir.

- **Adım 2:** Listenin ilk iki işi, önce ilk iş ardından ikinci iş ve önce ikinci iş sonra ilk iş olacak şekilde 2 türlü çizelgelenir. Yayılma zamanının düşük olduğu durumdaki atama sırası sabitlenir.
- **Adım 3:** 3 işten n işe kadar dördüncü adım tekrarlanır.
- **Adım 4:** k 'inci işin eklenmesi, k tane olası durumdan yayılma zamanı en küçük olana yapılır.

Bu sezgisel çözüm yöntemi permütasyon akış tipi çizelgeleme problemi için geliştirilmiş en etkili çözüm yöntemidir. Yapılan deneysel çalışma ile NEH algoritması ile hesaplanan yayılma zamanının $n = 5, \dots, 500$ ve $m = 5, \dots, 25$ için optimale ortalama olarak % 3 yaklaşık olduğu gösterilmiştir. Bu çalışma sonraki adım iyileştirmesi yapan bir çok sezgisel yöntem için temel teşkil etmektedir.

Endüstride sıklıkla kullanılan montaj hatları da akış tipi üretim sistemlerinin bir türüdür. 1900'li yılların başında Henry Ford tarafından kurulan ilk montaj hattı, günümüzde önemini korumaktadır. Temposu değişmeyen ve tek model üretilebilen hatlar Henry Ford'un ilk yıllarından itibaren geliştirilerek, paralel istasyonlu veya esnek işlemlerli hatlara, özelleştirilmiş karma ve karışık modelli hatlara, sabit kapasiteli ara stok bulunan belirli bir temposu bulunmayan U şeklindeki hatlara dönüşmüştür. Ancak bu dönüşüm, toplam iş yükünü hat boyunca üretim için montaj işleminin yapıldığı tüm istasyonlara dağıtılması sorunu olan montaj hattı dengeleme kavramını da beraberinde getirmiştir [7].

Bir montaj hattında bulunan k adet iş istasyonu birbirine konveyör gibi çeşitli malzeme taşıma sistemleri ile bağlı olabilir. Yarı mamul veya hammadde hattın bir ucundan girip, k adet istasyondan sırasıyla geçtikten sonra ürün haline dönüşür. İstasyonlarda yapılması gereken işlemlerin süreleri deterministik ya da stokastik olabilir. Montaj hattı dengeleme problemi bazı spesifik koşullar sağlanırken, hattaki toplam boş zaman miktarını enküçükleyecek şekilde işlerin sıralı iş istasyonlarına atanmasıdır. Bu koşullar; her bir iş istasyonuna atanan toplam görev zamanının, çevrim zamanına eşit ya da ondan küçük olması ve görevlerin atanmasında ardışık işlem sıralarının takip edilmesidir. Hat dengeleme istasyon sayısı sabit tutularak çevrim zamanının enküçüklenmesi veya çevrim zamanı sabit tutularak istasyon sayısının enküçüklenmesi şeklinde yapılır. Burada istasyon sayısının enküçüklenmesi istasyon maliyetini

enküçüklerken; çevrim zamanının enküçüklenmesi üretim hızını enbüyükler. Hattın durumuna, sistemden istenen performansa göre bu performans ölçütlerinden biri seçilerek dengeleme işlemi yapılır. Hat dengeleme problemi NP-Zor problem sınıfına girdiği için, büyük boyutlu problemler matematiksel model yardımıyla çözülememektedir. Literatürde hat dengeleme problemi için geliştirilmiş iyi sonuç veren oldukça fazla sezgisel yöntem vardır. Probleme konu olan montaj hattı iyi analiz edilerek, hattın etkinliğinin artırılması için mevcut yöntemler uygulanabilir ya da yeni yöntemler geliştirilebilir. Bu konuyla ilgili daha detaylı bilgi Becker ve Scholl [7] ve [8]'den elde edilebilir.

2.2 Üretimde Esneklik

1980'lerden günümüze literatürde üretimde esneklik konusunda bir çok çalışma yapılmıştır. Bu çalışmaların büyük bir kısmı, çeşitli esneklik tiplerini tanımlamak ve sistemleri bir veya birkaç esneklik tipi ile özdeşleştirmek üzerinedir. Bazı çalışmalar ise, çeşitli esnekliklerin faydalarını ve nasıl ölçülebileceklerini konu almaktadır. Yapılan bu çalışmalar esnekliğin karmaşık ve çok boyutlu bir konsept olduğunu göstermiştir ([9], [10], [11]).

Sistemin esnekliği, karşılaşılabilecek olası durumlara uyum yeteneğidir. Esnek bir sistem, değişken bir çevreye uyum sağlamalıdır. Üretimde esneklik, üretim kaynaklarının kaliteli ve farklı ürünler üretmek için şekillendirilmesidir [9]. Üretimde esneklik ayrıca değişen çevre koşullarına ve sistemde meydana gelen değişikliklere karşı bir tepki ve bu tepkinin gereksinimlerini karşılamak için kullanılan alternatif yollar olarak da tanımlanabilir [12].

Literatürde 50'nin üzerinde esneklik çeşidi tanımlanmış olmasına rağmen, bunların bir çoğu aynı şeyi ifade etmektedir. Sethi ve Sethi [9]'nin çok iyi bilinen çalışmasında esneklik tanımları incelenmiş ve sonuçta makine, malzeme taşıma, operasyon, süreç, ürün, rota, hacim, büyüme, program, üretim ve piyasa olmak üzere 11 ayrı kategoride sınıflandırılmıştır. Bu kategorilerden makine, malzeme taşıma, operasyon, süreç, ürün, rota ve büyüme esneklikleri sistemin önemli bileşenlerini; program, üretim ve piyasa esneklikleri ise üretim sisteminin tümüne uygulanan esneklikleri ifade etmektedir. Her bir esneklik türünün sisteme getirisi farklıdır. Esneklik türleri kısaca aşağıdaki şekilde

tanımlanabilir:

- **Makine Esnekliđi:** Bir makinenin deđişik tipteki operasyonları kolaylıkla yapabilecek kabiliyette olmasıdır.
- **Malzeme Taşıma Esnekliđi:** Farklı parça tiplerini uygun konumlara yerleştirme ve üretim tesisi boyunca işleme için etkin bir biçimde taşıma yeteneđidir.
- **Operasyon Esnekliđi:** Bir parçanın farklı şekillerde üretilebilmesidir.
- **Süreç Esnekliđi:** Bir üretim sisteminin temel kurulumları yapılmadan üreteceđi parça çeşitleri kümesi ile ilişkilidir.
- **Ürün Esnekliđi:** Mevcut parçalar için yerine koyma veya ekleme kolaylıđıdır. Üretilen farklı bir parçadan diđerine geçişin ucuz ve hızlı olmasını sađlayan ürün esnekliđidir.
- **Rota Esnekliđi:** Sistem boyunca farklı rotalar ile parça üretebilme yeteneđidir. Alternatif rotalar farklı makineler, farklı operasyonlar ya da farklı operasyonların sıralamaları kullanarak yapılabilir.
- **Hacim Esnekliđi:** Farklı çıktı seviyelerinde kârlı çalışma yeteneđidir.
- **Büyüme Esnekliđi:** Gerektiğinde kapasite artırılabilme kolaylıđıdır.
- **Program Esnekliđi:** Süreç ve rotalama esnekliklerine bađlı olup hazırlık zamanlarını azaltarak toplam üretim zamanının azaltılması, denetim ve ölçümünün geliştirilmesi ile ilgilidir.
- **Üretim Esnekliđi:** Temel sermaye ekipmanları eklenmeksizin üretim sisteminin ürettiđi parça çeşitleridir.
- **Market Esnekliđi:** Üretim sisteminin deđişen piyasa koşullarına uyum sađlama kolaylıđıdır.

Bu araştırmaya konu olan problem ile en çok ilgili olan esneklik türleri makine ve operasyon esnekliđidir. Bu iki esneklik türü Sethi ve Sethi [9]'ye göre daha detaylı olarak aşıđıdaki şekilde ifade edilebilir:

“Makine esnekliđi, bir makinenin bir operasyondan diđerine geđerken engelleyici bir kuvvet gerektirmeden deđişik tipteki operasyonları yapabilmesidir. Montaj hatlarında robotlar bu esnekliđi sađlarken, engelleyici kuvvet genelde zaman ya da maliyetle ifade edilir. Makine esnekliđinin teknolojik kaynakları numerik kontrol, kolay eriřilebilir programlama, sofistike parça yükleme ve alet deđiřtirme cihazları vb.dir. Çok amaçlı, çok eksenli CNC makineleri yüksek makine esnekliđine sahiptir.

Operasyon esnekliđi parçanın bazı operasyonlarının yer deđiřtirme veya deđiş tokuř edilmesi gibi alternatif süreç planları ile üretilebileceđi bir özelliktir. Bu esneklik çeřidi, rota esnekliđi gibi bir çok esnekliđe katkı sađlar. CAD/CAM, bilgisayarlı süreç planlama (CAPP) gibi sistemlerde ve grup teknolojisi operasyon esnekliđine sahip parçalar üretilmesini sađlar. Akıř tipi sistemlerde kaynaklar makineler ya da istasyonlardır.“

Ayrıca operasyon esnekliđi istasyon sayısından fazla iř olduđu ve iřlerin tümünün yapılacak şekilde istasyonlara atanabileceđi varsayımı altında, iřlerin istasyonlara atanması ile ilgilidir [13].

Mikro iřlemci teknolojilerinin ortaya çıkması ile esneklik, üretim sistemlerinin tasarım, operasyon ve yönetiminde kilit nokta haline gelmiştir. Esnek üretim sistemi için, olmazsa olmaz özellikler esneklik ve otomasyondur. Bununla birlikte otomasyonun derecesi ve parça çeřitliliđi de önemlidir. Bazı sistemlere sadece malzeme taşıma sistemi olduđu için esnek üretim sistemi diyemeyiz. Ayrıca entegre olmamıř sadece NC ya da CNC makinelerin bulunduđu sistemler ya da sadece bilgisayar kontrollü ama uzun hazırlık zamanları olan ve otomatik transfer hatları bulunmayan sistemler de esnek üretim sistemleri deđildir. Tüm esnek üretim sistemleri, büyük ölçüde çeřitli esneklik tiplerini kullanma potansiyeline sahiptir. Bazı ekonomik, teknolojik, pratik ve stratejik sebeplerden dolayı tüm esneklik tiplerini belirli bir esnek üretim sistemine uygulamak mümkün olmaz. Ayrıca otomasyon seviyesi de kullanılabilir esneklik seviyesini belirlemeye yardımcı olur [14].

Sistemin esnek olup olmadıđını, esnek ise hangi esnek üretim sistemi içinde sınıflandırılabilceđini belirlemek için takım tezgahları, malzeme taşıma sistemi, iřlem süreci içindeki stok alanları ve bilgisayarlı kontrol gibi çeřitli standartların sistemdeki durumunu incelemek gerekir. Browne vd. [14] esnek üretim sistemlerini ařađıdaki

gibi sınıflandırmıştır:

- **Esnek işlem hücresi:** Hammadde ve yarı mamül giriş stoğundan alarak makinelere getiren, makinelerde yükleme ve boşaltma işlemlerini yapan ve sonunda çıkış stoğuna getiren otomatik malzeme taşıma sistemini içerir. Burada çoğu zaman genel amaçlı NC ve CNC makineler kullanılır. Makine yükleme ve boşaltma işlemleri için zaman zaman robot, çeşitli taşıyıcı kollar veya palet değiştirici kullanılabilir.
- **Esnek işleme sistemi:** Üretimde gerçek zamanlı çevrimiçi kontrol sistemi ile takip edilen düşük hacimli parça üretimi ve parçaların bir çok rota ile işlenebildiği genel amaçlı takım tezgahlarının bulunduğu sistemlerdir. Bu esnek üretim sistemi yüksek makine, süreç, rota ve ürün esnekliğine sahiptir.
- **Esnek transfer hattı:** Bu sistemlerde tüm parça çeşitleri için her bir operasyonun yapılabildiği tek bir makine vardır. Bu yüzden tüm rotalar sabittir. Malzeme taşıma sistemi olarak genellikle konveyörler kullanılır. Genel amaçlı makinelerin yanısıra robotlar veya özel bir iş için tahsis edilmiş bazı özel amaçlı makineler de bulunabilir. Bilgisayarlı kontrol sistemi daha basittir. Bu sistem esnek işleme sistemine göre daha az süreç esnekliğine sahip olmakla birlikte, tekrar işleme, hızlı hazırlık süreci ve bilgisayara manuel komutlar verme özelliklerine sahiptir.
- **Çoklu transfer hattı:** Esnek işleme sistemi ve esnek transfer hatlarının, en iyi özelliklerinden faydalanmak için, birbirlerine bağlanmaları ve beraber kullanılmaları ile elde edilir. Özellikle makine arızalarında yüksek rotalama esnekliğinin faydasını kullanır.

Mevcut bir sisteme esneklik adapte edilmesi ya da esnek bir üretim sistemi kurulması kolay değildir. Hangi esneklik türünün uygun olduğu ve bunun için ne kadar yatırım yapılması gerektiği oldukça zor kararlardır. Diğer taraftan esnekliğin sisteme kazandıracağı faydalar net bir şekilde belirtilemezlerse firmalar esneklik için yatırım yapma konusunda direnç göstermektedirler [15]. Buna paralel olarak Jordan ve Graves [16] yaptıkları çalışmada, esnekliğin nasıl ve ne kadar uygulanması gerektiği ile ilgili güzel yaklaşımlarda bulunmuşlardır. Buna göre, imalat işlemlerinin bir çoğunun esnek olmaması bir çok problem ortaya çıkmasına neden olur. İlk olarak,

kapasite ve esneklik planlamasında, esnek operasyonların yatırım maliyeti hesaplanır. Buna rağmen esnekliğin yararları genellikle ölçülemez. Çünkü talep belirsizliği planlamacılar tarafından gerçek bir şekilde hesaba katılamaz. Esneklik pahalı olduğu düşünüldüğü için de yatırım kararı alınmaz. İkinci bir problem, endüstride çalışan veya akademik bir çok kişinin esneklik uygulandığı takdirde ya hep ya hiç olacağı görüşünde olmalarıdır. Bu durum da, ilk maddede açıklanan esneklik maliyetinin çok olduğu düşüncesiyle genellikle esnekliğin uygulanmaması kararı ile sonuçlanır [16].

Esneklik türlerinin faydalarını araştıran çeşitli çalışmalar vardır. Örneğin Jordan ve Graves [16] süreç esnekliği varsayımı altında incelediği sistemde satış kayıplarının % 89 oranında azaltılabileceğini göstermişlerdir. Daniels ve Mazzola [17], az bir esnekleştirme maliyeti ile fazladan esnekliğin sağlayabileceği azami faydanın büyük bir kısmının elde edilebileceğini göstermişlerdir. Bu çalışmada yazarlar, esneklikten mümkün olan en çok faydayı sağlamak için esneklik miktarı ve çizelgeleme gibi operasyonel kararların birlikte ele alınmaları gerektiğini ifade etmiştir. Benzer çıkarımlar Nomden ve van der Zee [18] tarafından da yapılmıştır. Firmaların esneklikten fayda sağlamak için düşündükleri kadar yüksek miktarlarda yatırım yapmalarına gerek olmadığını, düşük maliyetlere katlanarak yüksek kazançlar elde edebileceklerini göstermektedir.

2.3 Esnek Akış Tipi Üretim Sistemlerinde Çizelgeleme

Bu bölümde, bu çalışmanın konusu olan problemle en yakından ilgisi olan çalışmalar ve bu çalışmalarda elde edilen sonuçlar detaylı bir şekilde incelenecektir.

Esnek akış tipi üretim sistemlerinde, makine ya da operasyon esneklikleri varsayımı altında geleneksel akış tipi sistemlerden farklı olarak bazı işlerin birden fazla makine ya da işçi tarafından yapılabilmesi özelliği bulunmaktadır. Bu sayede iş yükleri daha dengeli dağıtılabilir, işlerin tamamlanma zamanı azaltılarak toplamda üretilen miktar artırılabilir.

Akış tipi üretim sistemi çizelgeleme konusunda literatürde çok sayıda çalışma bulunurken, esnek akış tipi üretim sistemlerindeki yayın sayısı ancak son yıllarda artmaya başlamıştır. Problemle yakından ilişkili yayınlarda genellikle sezgisel algoritmalar geliştirilmiş ve optimale yakın sonuçlar bulunmuştur.

Daniels ve Mazzola [19], n adet iş ve m adet makineden oluşan, kaynakların esnek olduğu ve dinamik olarak atanabildiği bir çizelgeleme problemini ele almıştır. Problem çözümünü kolaylaştırmak için tek bir kaynak ele alınmıştır. Bu kaynak çapraz eğitilmiş işçi ya da esnek makine olabilmektedir. Amaç, iş sıralamasını belirlemek, her bir işlem için kaynak atama kuralını bulmak ve yayılma zamanını enküçüklemektir. Problemin karmaşıklığı tartışılmış, optimal çizelgeler için tanımlar belirlenmiştir. Sezgisel ve optimal çözüm yaklaşımları geliştirilmiştir. Kaynak esnekliğinin operasyonel yararlarının kavranması için makine sayısının 2, 3 ve 4 olduğu, 10, 15 ve 20 işlemden geçerek tamamlanan 5'er parçanın üretildiği sistemler ele alınarak geniş kapsamlı bir deneysel çalışma tasarlanmıştır. Analizler sonucunda esneklik dahil olan sistemin sezgisel çözüm yaklaşımı ile bulunan yayılma zamanının, normal bir akış tipi üretim sistemine göre ortalamada % 12,2 daha iyi olduğu görülmüştür. 100 kadar işlem içeren problemde kaynak esnekliğinin yayılma zamanına olan etkisi ayrı ayrı gözlemlenmiş, performans gelişmeleri % 81 olarak hesaplanmıştır. Performansta meydana gelen iyileşmeler kaynak esnekliğinin önemli olduğunu ve geliştirilmiş sezgisel yöntemin büyük boyutlu problemler için etkili sonuçlar elde ettiğini gösterilmiştir.

Daniels ve Mazzola [17], n adet iş ve m adet makineden oluşan bir sistemde işçi esnekliğine odaklanarak, w adet işçi çizelgelemesini konu almıştır. Çalışmanın amacı, kısmi kaynak esnekliği olarak ifade edilen, çapraz eğitilmiş işçilerin birbirleri yerine işlemler yapabilmeleri durumunun akış tipi sistemlerdeki operasyonel faydalarını keşfetmektir. Kısmi kaynak esnekliği için metrikler belirlenmiş ve kabiliyet matrisleri oluşturulmuştur. Buradan hareketle kısmi kaynak esnekliği olan bir akış tipi sistemin optimal bir şekilde çizelgelenmesi için dal-sınır algoritması geliştirilmiştir. 5 iş-3 istasyon ile 4 iş-4 istasyon durumlarını aşan problemlerde optimal çözüm bulmak zor olduğu için sezgisel bir algoritma geliştirmişlerdir. Yapılan geniş kapsamlı deneysel çalışma ile işçi esnekliğinin faydasının büyük bir kısmının, daha az bir çapraz eğitim maliyeti ile gerçekleştirilebileceği gösterilmiştir.

Burdett ve Kozan [20], sıfır ara stok kapasitesine sahip, karma modelli bir sistemde iş sıralama ve çizelgeleme problemini ele almıştır. Ara stok olmamasından dolayı istasyonlar bloke olabilmekte, bu durum da istasyonlarda boş zamanlar açığa çıkarmaktadır. Bu çalışma, bu bloke neticesinde açığa çıkan boş zamanları azaltmak ya da ortadan kaldırmak ve çok becerili işçilerin kullanımını artırmak amacıyla yapılmıştır. Sabit ara stok kapasitesine sahip, beklemez sistemler de incelenmiştir. İşlerin kaydırılması (veya değiştirilmesi) ve yeniden düzenlenmesi metotları uygulanmıştır.

Tüm kaydırma senaryoları incelenmiş ve bir matematiksel model geliştirilmiştir. Ancak matematiksel modelin büyüklüğü ve karmaşıklığından dolayı çözüme ulaşmak için tavlama benzetimi kullanılmıştır. İş sıralama ve çizelgeleme için bir çok farklı test yapılmıştır. Yapılan testler neticesinde çeşitli senaryolar altında, işler ve istasyonların yeniden düzenlenmesi ile yayılma zamanı azaltılabileceği gösterilmiştir. Ayrıca bu iyileşmenin sabit, yani parça ve istasyon sayısından bağımsız, olduğu görülmüştür. Özellikle ara stok kapasitesi olmayan sistemlerde paylaşımlı işlerin bulunduğu senaryonun büyük yararları olduğu gösterilmiştir.

Gupta vd. [21], n adet birbirinden farklı parça, her parça üzerinde yapılması gereken 3 işlem ve 2 makine bulunan bir sistemde tamamlanma zamanını enküçüklemek için, her bir parçanın esnek işleminin hangi makinede yapılacağını bulmaya yönelik bir çalışma yapmışlardır. Çalışmanın amacı, yayılma zamanını enküçüktür. İlk iş ilk makinede, üçüncü iş diğer makinede yapılmak zorundadır. İkinci iş ise ilk ya da ikinci makinelerin herhangi birinde yapılabilir. Gözlemler sonucunda çeşitli önteoremler belirlenmiş ve ispatları yapılmıştır. Problemin NP-Zor olduğu gösterilmiştir. Bu yüzden, önteoremlerden yola çıkılarak yaklaşım algoritmaları geliştirilmiştir. Esnek işlemlerin atamasını keyfi olarak yapan ilk algoritmanın kötü durum performansı 2'dir. Yani algoritma kullanılarak elde edilen yayılma zamanı en kötü durumda optimal yayılma zamanının iki katı değer almaktadır. Yazarlarca geliştirilen ve 4 adet çizelge oluşturarak bunlardan en iyisini seçen algoritmanın ise kötü durum performansı 3/2'dir. Polinom zamanlı bu algoritma, herhangi bir zamanda en az bir makinenin boş zamanı olmayacağını garanti eder. Bu çalışmada ele alınan problem türü, tez çalışması ile aynı esneklik karakteristiklerini taşımaktadır. Esnek makinelerin özellikleri, esnek işlem yapabilme türleri, problem tanımı sırasında verilen örnekler, tez çalışmasına birebir uyumludur. Tez çalışmasından farkı ise, bu tezde m adet makinenin ele alınması ve parçaların tek tip olmasıdır.

Askin ve Chen [22], 2 makineli, her bir işin tek bir iş istasyonuna atanabileceği, iş sıralarının belirli olduğu bir sistem ele almıştır. Bazı işler, kısmi çapraz eğitimli işçiler tarafından, birden çok istasyonda yapılabilir. Çapraz eğitim ile oluşan esnekliğin hatta denge sağlaması ya da hat dengesini artırması beklenmektedir. Sistemde var olan stok maliyeti ile çapraz eğitim maliyetini dengeleme amacı vardır. İşlem zamanları rastgele olarak seçilmiştir ve küçük miktarda ara stok bulunabilir. Ardışık istasyonlarda farklı seviyelerde çapraz eğitime sahip çeşitli işçiler için çeşitli kurallar belirlenmiştir.

Bu kurallar ile Arena kullanılarak bir benzetim modeli oluşturulmuş ve Arena Opt-Quest ile elde edilen değerler ile kıyaslanmıştır. Daha önce belirlenen kuralların hangi modeller için hangi durumlarda gerçekleştiği benzetim modeli ile incelenerek gözlemler belirlenmiştir. Buna göre çapraz eğitilmiş işçilerin daha az ara stok miktarları ile daha yüksek çıktı oranlarına ulaştıkları gözlenmiştir.

Anuar ve Bukchin [23], klasik bir akış tipi üretim sistemine esneklik eklenerek, çevrim zamanı kısaltılmak istenmiştir. Öncelikle dengelenebilir hatların özelliği belirlenmiş, hat dengeleme ile elde edilen başlangıç ataması analiz edilmiş, paylaşımlı işler ve durumla bağımlı ve bağımsız operasyonel kurallar belirlenmiştir. Daha sonra dinamik hat dengeleme yöntemiyle hesaplanmış mevcut durumdaki en iyi denge, çevrim zamanını minimize edecek şekilde geliştirilmiştir. Paylaşımlı iş kullanma maliyeti belirlenmiş, paylaşım zamanını azaltmak ve kârı enbüyüklemek için bazı hızlı optimal algoritmalar önerilmiştir. Hattın farklı makine sayısı ve ara stok alanı gibi özellikleri için kapsamlı bir deneysel çalışma yapılmıştır. Paylaşımlı işler için önerilen yaklaşım, klasik montaj hattı yaklaşımına göre çevrim zamanında azalma sağlamıştır. Özellikle ardışık istasyonlar arasında düşük ara stok seviyelerinde belirtilmiş olan bazı operasyonel kuralların çevrim zamanında iyileşme sağladığı görülmüştür.

Crama ve Gültekin [24], 2 makineli bir akış tipi üretim sistemini ele almıştır. Aynı tip parçaların üretildiği bu sistemde, esnek makinelerin farklı işler yapabildikleri varsayımı vardır. Ancak çeşitli kısıtlardan dolayı bazı işler sadece belli makinelerde yapılmakta olup, bazıları birden çok makinede de yapılabilmektedir. Bu durumda, ilk iş sadece ilk makinede, ikinci iş sadece ikinci makinede yapılabilmektedir. Esnek işlem ise her iki makinede de yapılabilmektedir. Problem, çıktı miktarını enbüyükleyecek şekilde her bir parça için esnek işlemin hangi makinede yapılacağı kararını vermektir. Parça sayısının n veya sonsuz; makineler arası stok kapasitesinin sıfır, sabit kapasiteli veya sonsuz olduğu durumlar farklı problemler olarak değerlendirilmiştir. Ele alınan bu problemler için optimal çözümün çeşitli özellikleri belirlenmiştir. Bu sonuçlardan yola çıkılarak farklı parça sayısı ve ara stok kapasiteleri ile ayrıştırılmış olan problem türleri için optimal çözümü veren polinom zamanlı çözüm yöntemleri geliştirilmiştir. Bu çözüm yönteminde, her bir parça atanırken iki parça sonrasının atmasına da bakarak, ileriye doğru adım adım bir iyileştirme sağlanmaktadır. Bu çalışma, esnek makinelerin ardışık olması, esnek ve sabit işlem sürelerinin parçadan parçaya değişmemesi, aynı tip parçaların ele alınmış olması, ara stok kapasitelerinin sıfır veya sınırsız olması gibi özellikler nedeniyle mevcut tez çalışması ile benzer özellikler

göstermektedir.

2.4 Özet

Literatür araştırması bölümünde, geleneksel akış tipi çizelgeleme problemleri, üretimde esneklik ve esnek akış tipi üretim sistemlerinde çizelgeleme konularında yapılmış çalışmalar incelenmiştir.

Bu tez çalışmasına konu olan problem özellikle Crama ve Gültekin [24], Gupta vd.[21] ve Daniels Mazzola [17]'nin yapmış oldukları çalışmalar ile problem tanımı açısından benzer özellikler göstermektedir. Problem konusu, Crama ve Gültekin [24] ile Gupta vd. [21]'nin m makine için uygulanmış halidir. Gupta vd. [21]'deki 2 makine için yapılan çalışmadan hareketle, Crama ve Gültekin [24] problemi aynı tip parça için çok daha çeşitli alt problemler halinde incelemiş ve optimal sonuçları bulmuştur. Bu tez çalışması, problemi m makine için genişletmiş ve daha önce yapılmış çalışmalarını dikkate alarak yeni çözüm yöntemi arayışlarına girmiştir.

Düşük makine sayısı olan problemler için geçerli olan çözüm yöntemlerinin, makine sayısı artırılarak gerçek hayatta karşılaşılan modellere daha çok benzetildiği durumlarda da geçerli olup olmadıkları bir soru işaretidir. Bu çalışmanın amacı, literatürde şimdiye kadar ele alınmış çalışmalarını genişletmek ve gerçekçi hale getirmektir. İlerleyen bölümlerde de görüleceği üzere, önceki çözüm yöntemleri makine sayısı artırıldığında kullanılmamaktadır. Bunun sonucu olarak bu çalışmada yeni çözüm yöntemleri geliştirilmiştir. Bu açıdan önemli bir boşluk doldurulmuştur.

Görüldüğü üzere, giderek değişen müşteri taleplerini karşılayabilmek ve rekabetçi üretim endüstrisinde varlığını devam ettirebilmek için esnek üretim sistemlerinin kullanımı giderek yaygınlaşmaktadır. Endüstrideki uygulama alanlarının oldukça geniş olduğu ve sıkça karşılaşıldığı ifade edilmiş olan bu problemin çözülmesinin getireceği kazançlar açıktır. Ancak bu çalışmada ele alınan problem daha önce hiçbir çalışmanın konusu olmamıştır. Bir çok üretim sisteminde bulunan m -makinelikli akış tipi sistemler ve her bir makinenin yapabileceği değişik işlemlerin olduğu en genel problemlerin çözümü ve zorluğu için öngörüler elde etmek adına bu çalışma önem taşımaktadır.

3. Problem Tanımı ve Eniyileme Modeli

Bu bölümde, çalışmaya konu olan problem detaylı bir şekilde tanımlanmıştır. Problemin endüstride karşılaşıldığı uygulama alanları, ele alınan esnekliğin sisteme getirisinin ne olduğu ve problemin zorluğu ile ilgili bilgiler verilmektedir. Ayrıca problemin sınırsız veya sıfır ara stok kapasiteleri için kurulmuş eniyileme modeli de bu bölümde sunulmaktadır.

3.1 Problem Tanımı

Bu çalışmada $m > 2$ makineli bir akış tipi üretim sistemi ele alınmıştır. n adet aynı tip parça, işlemlerinin tamamlanması için tüm makinelerden sırayla geçmek zorundadır. Her makinenin yapması gereken sabit bir işlem vardır. Bununla birlikte, parçanın gereksinim duyduğu işlemlerden bir tanesi de sabit işlemlere ek olarak k 'inci veya $k + 1$ 'inci sırada bulunan makine tarafından yapılabilen bir işlemdir. Bu çalışma boyunca, bu işleme “esnek işlem”; bu işlemi yapabilen k ve $k + 1$ numaralı makinelere ise “esnek makineler” adı verilecektir. Dolayısıyla, bir parçanın işleminin tamamlanması için toplamda m adet makinede yapılacak $m + 1$ işlemin tamamlanmış olması gerekmektedir. Yayılma zamanı (makespan) ise n 'inci parçanın m 'inci makinedeki tamamlanma zamanına eşittir. m makineli sistemin genel durumu Şekil 3.1'de görülmektedir.

Esnek işlem, her bir parça için sadece bir esnek makinede yapılabilir. Bu yüzden üretilecek her parça için esnek işlemin hangi makinede yapılacağına karar verilmesi gerekir. Hattaki her bir makine arasında belirli kapasiteli bir stok alanı bulunabilir.

eđitimli iřçilerin çalıřtıđı montaj hatları ([17], [19], [20], [22], [23]) ve PCB kart üzerine devre elemanlarının yerleřtirildiđi elektronik devre üreten sistemler ([25]) olarak ifade edilebilir.

İdeal bir esnek üretim sisteminde CNC makineler, takım depolarında gerekli takımlar olduđu sürece gerekli iřleri yapabilirler [26]. Fakat, takım depoları sınırlı kapasiteye sahiptir ve bir parçayı üretmek için gerekli bütün uçlar genellikle takım depolarına sığmazlar. Ayrıca, CNC makinelerinde kullanılan bazı uçlar çok pahalıdır ve bu uçlardan her makineye birer kopya yerleřtirmek ekonomik olarak mümkün deđildir. Bu nedenlerle, bazı iřleri yapan uçlar birden çok makinede bulunurken, bazı uçlar sadece belli makinelerde bulunabilmektedir. Aynı türden uçlar makinelerde esnek iřlemlerin yapılmasını sađlarken, makinelere has uçlar sabit iřlemleri belirlemektedir.

Monotonluđu azaltmak ve iřçilerin birbirleri yerine de çalıřabilmelerini sađlayabilmek gibi nedenlerle yapılan çapraz eđitim, üretim sistemlerine esneklik katmaktadır. Burada da iřçilerin sadece kendilerinin yapabildikleri iřler sabit iřleri oluřtururken sistemin durumuna göre deđiřtirilebilecek paylařımlı iřler esnek iřlemleri oluřturmaktadır. Burada iřçilerin çapraz eđitime tabi tutulmasının yanı sıra, iřlemi gerçekleřtirmek için gerekli aletlerin de çođaltılması ve farklı istasyonlara yerleřtirilmesi gerekmektedir.

PCB kart üzerine devre elemanlarının yerleřtirildiđi elektronik devre üreten sistemlerde, devre elemanlarını yerleřtiren robot kollara her bir elemanı monte edebilmeleri için tutucular yerleřtirilmiřtir. Her robot kol en fazla belli sayıda tutucuyu üzerinde bulundurabilmektedir. Fakat tutucuların fiziksel büyüklükleri farklıdır. Dolayısıyla her bir tutucudan bütün robot kollara yerleřtirmek mümkün deđildir. Her bir robot kolda farklı olarak bulunan tutucular sabit iřlemleri; aynı tutucular ise esnek iřlemleri gerçekleřtirmektedir.

Gupta [21], esnek iřlemlere örnek olarak bir stok kontrol noktasında yapılan iřlemleri vermiřtir. Farklı kaynaklardan gelen maddelerin kutularını açma, stok kontrol için barkod okuma ve tekrar paketleyip önceden tanımlı alıř noktalarına gönderme yapılacak iřlemlerdir. Barkod okuma iřlemi otomatikleřtirilmiř bir sistemle yapıldıđı için hem kutu açma hem de paketleme istasyonlarında yapılabilmektedir. Bu yüzden barkod okuma iřlemi esnek iřlem; kutu açma ve yeniden gruplayıp paketleme iřlemleri de sabit iřlemler olarak ifade edilebilir.

Esnek işlem Néron vd. [27] tarafından çiftçilik alanında bir örnekle tanımlanmıştır. Ekim alanları sırayla temizleme, sürülme ve ekim işlemlerine tabi tutulmaktadır. Bu işlemlerden temizleme ve bazı sürülme işlemleri erkekler; bazı sürülme işlemleri ve ekim kadınlar tarafından yapılmaktadır. Bu örnekte erkek ve kadın grupları için ortak yapılabilen sürülme işlemi esnek işlem; temizleme ve ekim işlemi ise sabit işlemlerdir.

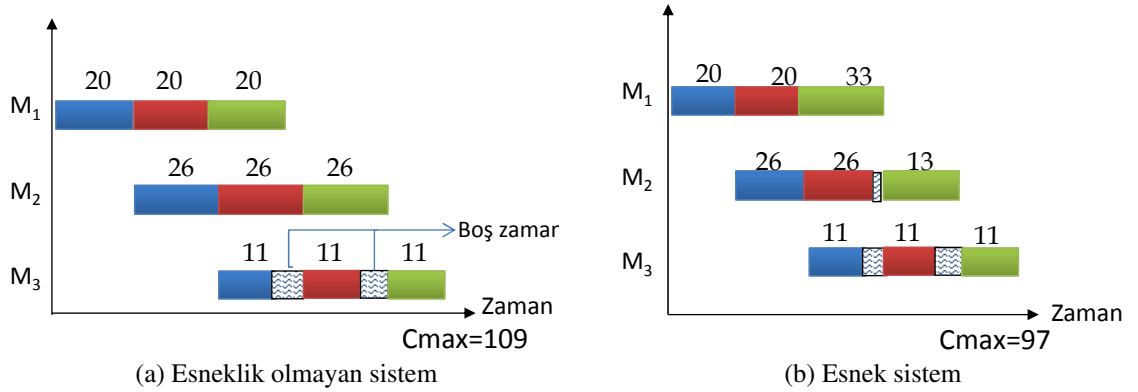
3.3 Esnekliğin Getirisi

Önceki bölümlerde bahsedildiği gibi esnek üretim sistemlerinin kullanımı son yıllarda giderek yaygınlaşmakta ve bu konuda yapılan bilimsel çalışmaların sayısı giderek artmaktadır. Esnekliğin sisteme getirdiği faydaların araştırılması ve belirlenmesi bu sistemlerin neden yaygınlaştıklarını kolayca ortaya koyabilmektedir. Aşağıdaki örnek, bu çalışmada ele alınan sistem için, esnekliğin üretim hızına getirisini ortaya koyması açısından önemlidir.

Örnek 1 $n = 3, m = 3, k = 1$ olan ve esnekliğin CNC makinelerle sağlandığı bir akış tipi üretim sisteminde sabit işlem süreleri sırasıyla 20, 13, 11 ve esnek işlem süresi 13 olsun.

Şekil 3.2a'da verilen Gantt şemasında esnekliğin olmadığı bir üretim sisteminde elde edilecek en düşük yayılma zamanını gösterilmektedir. Bu durumda en iyi çözümü elde edebilmek için tüm işler sabit işlem süresi küçük olan 2'nci makinede gerçekleştirilmektedir. Bu durumda 2'nci makinenin işlem süresi 26'ya yükselmiş, 1'inci makineye hiç bir esnek işlem ataması yapılmadığı için işlem süresi 20 olarak kalmıştır. Bu atama sonucunda, yayılma zamanı 109 olarak hesaplanmıştır.

Şekil 3.2b'de ise esneklik dahil olan bir sistemin optimal atama durumu görülmektedir. İlk iki işin esnek işlemi ikinci makineye atandığından ikinci makinenin işlem süresi bu parçalar için 26'ya çıkmış; son işin esnek işlemi birinci makineye atandığı için ise birinci makinenin işlem süresi bu parça için 33 olmuştur. Bu atama ile hesaplanan yayılma zamanı 97'dir. Ele alınan örnekte, esneklik dahil olan sistem ile klasik montaj hattı dengeleme sonucuna göre % 11 kazanç elde edilmiştir. Görüldüğü üzere bu küçük örnekte bile ciddi bir kazanç sağlanmıştır. Kazancın çok daha fazla olacağı örnekler türetmek de mümkündür. Diğer taraftan, bu kazanç sadece küçük bir esnekleştirme



Şekil 3.2: Örnek 1 için esnek ve esnek olmayan sistemlerin optimal Gantt şemaları

maliyeti ile elde edilebilecek bir kazançtır. Tek yapılması gereken esnekleştirilmek istenen işlem için gerekli olan kesici uçtan bir tane daha alınması ve her iki makineye de yerleştirilmesidir. Bu noktada önemli olan, esnekleştirilmiş sistemde her parça için esnek işlemin hangi makineye atanması gerektiğinin belirlenmesidir.

3.4 Problemin Zorluğu

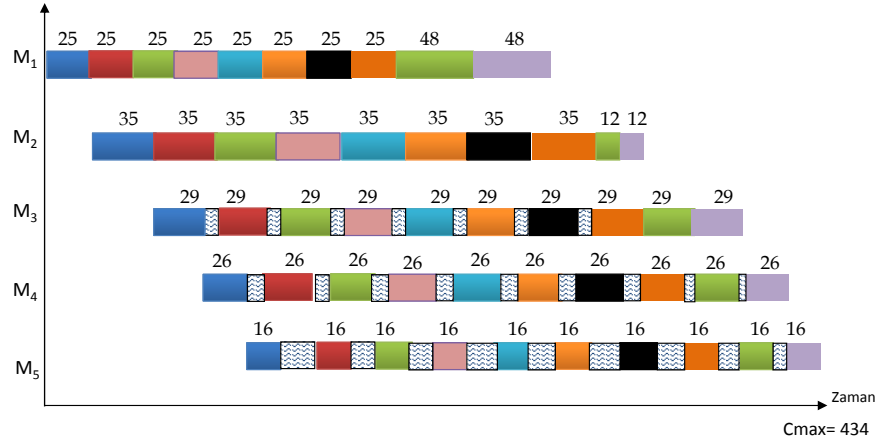
Bir çok çizelgeleme problemi, özellikle gerçek hayatta karşılaşıldığı büyüklükleri ile, tamsayılı programlama gibi matematiksel programlama modelleri aracılığı ile kolayca çözülemeyecek kadar karmaşıktır. Bu yüzden dal-sınır ya da dinamik programlama gibi klasik teknikleri uygulamak da zor olmaktadır. Bu problemler çoğu zaman gerçek zamanlı olarak çözülemezler [28].

Bu çalışmaya konu olan probleme en yakın çalışma, Crama ve Gültekin [24] tarafından yapılmıştır. Ayrıca Gupta vd. [21] n adet birbirinden farklı parça üreten problemi ele almıştır. Bu problemde işlemlerin sabit zamanları sıfır olsa bile geriye kalan problem, NP-Zor olduğu bilinen, 2-paralel makinede yayılma zamanının enküçüklenmesi problemidir. Gupta vd. [21]'de ele alınan problem farklı tip parça değil de aynı tip parça için geliştirilmiş olsaydı, NP-Zor olarak sınıflandırılmayacaktı. Çünkü bu problem, Crama ve Gültekin [24] tarafından polinom zamanlı bir algoritma ile çözülmüştür.

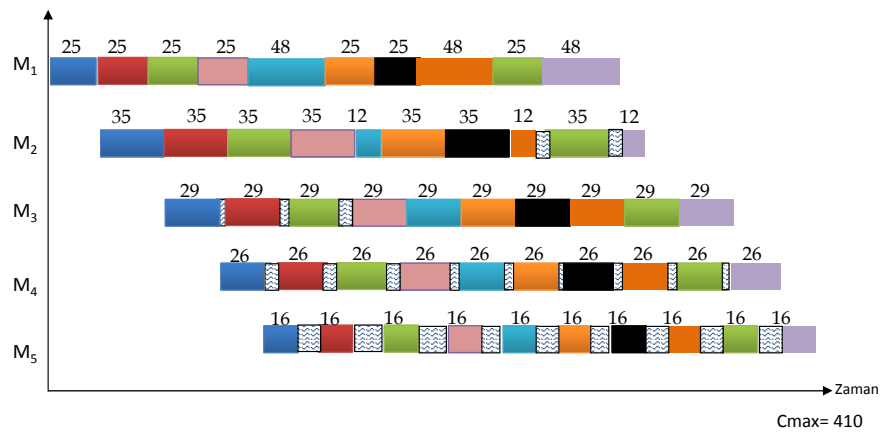
Crama ve Gültekin [24]'in sınırsız ara stok kapasitesine sahip 2 makineli sistemler için bir algoritma geliştirmiştir. Buna göre esnek işlemi ilk makineye atanacak parça

sayısını ifade eden ve r ile gösterilmiş bir değer belirlenmiştir. Bu değer $r = ((n - 1)(f^2 + s - f^1) + s)/(2s)$ formülü ile hesaplanmıştır. Hesaplanan değer tamsayı çıkmaması halinde üstteki en küçük tamsayı değeri $\lceil r \rceil$ ve altta bulunan en büyük tamsayı değeri $\lfloor r \rfloor$ kullanılmıştır. Hesaplanan değerler $C_{max} = \min\{f^1 + nf^2 + (n - \lfloor r \rfloor * s), nf^1 + f^2 + \lceil r \rceil s\}$ formülünde yerine konularak optimal yayılma zamanı elde edilmiştir. Buna göre en küçük yayılma zamanını veren $\lceil r \rceil$ veya $\lfloor r \rfloor$ değeri r olarak tutulmuştur. Bu değere göre ilk $n - r$ parça ikinci esnek makineye; sonraki r parça ilk esnek makineye atanarak optimal çözüm bulunmuştur.

Aşağıdaki örnek aynı çözüm yönteminin, k ve $k + 1$ 'inci makinelerin esnek makineler olduğu m makineli bir sistemde optimal çözümü vermeyebileceğini göstermektedir.



a- r hesabı ile ulaşılan çözümün Gantt şeması



b- optimal çözümün Gantt şeması

Şekil 3.3: Örnek 2 için r hesabı ve optimal çözümün Gantt şemaları

Örnek 2 5 makineli, 10 parçalı, sabit işlem sürelerinin sırasıyla 25, 12, 29, 26, 16 ve esnek işlem süresinin 23, esnek makinelerin 1 ve 2'nci makineler ($k = 1$) olduğu

bir sistemde deęerler formüllerde yerine konulduğunda $r = 2$ bulunur. Buradan hareketle, Şekil 3.3a'daki Gantt şemasında görüldüğü gibi, Crama ve Gültekin [24]'in önerdiği şekilde ilk 8 parçanın ikinci makineye, son 2 parçanın da ilk makineye atanması sonucuna ulaşılmış ve yayılma zamanı 434 olarak hesaplanmıştır. Ancak Şekil 3.3b'deki Gantt şemasında da görüldüğü gibi problemin optimal çözümünde 5'inci, 8'inci ve 10'uncu parçaların esnek işlemleri ilk makineye atanmıştır. Yani formülle hesaplanmış olan $r = 2$ yerine 3 parça atanmış ve yayılma zamanı 410 olarak hesaplanmıştır. Dolayısıyla hesaplanan r deęeri kullanılamamıştır. Diğer taraftan bu çalışmanın önemli bir özellięi olan ilk $n - r$ parçanın ikinci makineye, sonraki r parçanın da ilk makineye atanması kuralı da gerçekleşmemiştir. Bu örnek, daha önce 2 makineli sistemler için geliştirilmiş yöntemlerin m -makineli sistemlere aynen uygulanamadığını ve yeni bir çözüm yönteminin geliştirilmesi gerektiğini göstermektedir.

3.5 Eniyileme Modeli

Bu çalışma süresince kullanılacak parametreler aşağıdaki şekildedir:

n : Üretilecek parça sayısı

m : Sistemdeki makine sayısı

k : Esnek işlem yapan ilk makinenin sıra numarası, dolayısıyla esnek işlem yapan ikinci makine $(k + 1)$ ile adlandırılır.

f^j : j makinesindeki sabit işlem süresi

s : Esnek işlem süresi

Modelde kullanılan deęişkenler ise şu şekildedir:

T_i^j : i işinin j makinesindeki başlama zamanı

C_i^j : i işinin j makinesindeki tamamlanma zamanı. Bu tamamlanma zamanı, sınırsız ara stok kapasiteli sistemler için parçanın makineyi terk etme zamanına eşitken; sıfır ara stok kapasiteli sistemlerde eşit olmayabilir.

$$x_i^j = \begin{cases} 1; & i \text{ parçasının esnek işlemi } j \text{ makinesine atanmışsa } (j = k, (k + 1)) \\ 0; & \text{Değilse} \end{cases}$$

Herhangi bir esnek parçanın esnek işlemi, esnek makinelerden sadece bir tanesine atanmalıdır. Bu durum, eniyileme modelinde bulunan ve bir sonraki bölümde anlatılacak olan $x_i^k + x_i^{k+1} = 1$ kısıtı ile sağlanmaktadır.

Karar değişkenlerinin aldıkları değerlere göre, tamamlanma zamanı hesaplarında çeşitli düzenlemeler yapılmaktadır. Eğer j makinesi esnek makine değilse, i parçasının j makinesindeki işleme süresine esnek işlem süresi dahil edilmez ve tamamlanma zamanı, $C_i^j = T_i^j + f^j, \forall i, j \neq k, (k + 1)$ olarak hesaplanır. Eğer j makinesi esnek makine ise, i parçasının j makinesindeki işleme süresine esnek işlem süresi de dahil edilir ve tamamlanma zamanı hesabı, $C_i^j = T_i^j + f^j + s.x_i^j, \forall i, j = k, (k + 1)$ şeklinde yapılır.

Esnek işlem süresi s , atamaya göre sadece esnek makinelerin sabit işlem sürelerine eklenmektedir. İlk parça ilk makinede sıfır anında işlenmeye başlamaktadır. Sonraki parçaların başlangıç ve tamamlanma zamanları esnek makinelerin hangi makineler olduğuna, yapılan atamaya, esnek ve sabit işlem sürelerine göre hesaplanmaktadır.

Bu durumda, bir önceki bölümde çözülmüş olan Örnek 1, bu notasyonlar kullanılarak $n = 3, m = 3, k = 1, f^1 = 20, f^2 = 13, f^3 = 11$ şeklinde ifade edilebilir. Elde edilen çözüm ise, $i = 1, \dots, n, j = 1$ için $x_i^j = \{0, 1, 1, 1, 0, 1, 1, 0, 1, 1\}, T_1^1 = 0, T_1^2 = 20, T_1^3 = 46, \dots, T_3^3 = 97, C_3^3 = 410$ şeklindedir. C_3^3 aynı zamanda yayılma zamanını ifade etmektedir. $x_i^1 + x_i^2 = 1$ olduğu için x_i^2 'nin ayrıca verilmesine gerek yoktur.

Araştırmada ele alınan sıfır veya sınırsız ara stok durumları için optimizasyon modeli oluşturulmuştur. Aşağıdaki model sıfır ara stok durumu için düzenlenmiş olup, sınırsız ara stok problemi için (3.4) ve (3.5) kısıtlarının kaldırılması yeterli olmaktadır.

$$\min T_n^m + f^m \quad (3.1)$$

$$T_i^j \geq T_{i-1}^j + f^j \quad \forall i \neq 1, \quad \forall j \neq k, (k + 1) \quad (3.2)$$

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

$$T_i^j \geq T_{i-2}^{j+1} + f^{j+1} \quad \forall i \neq 1, 2, \quad \forall j \neq (k-1), k, m \quad (3.4)$$

$$T_i^j \geq T_{i-2}^{j+1} + f^{j+1} + s.x_{i-2}^{j+1} \quad \forall i \neq 1, 2, \quad j = (k-1), k \quad (3.5)$$

$$T_i^j \geq T_i^{j-1} + f^{j-1} \quad \forall i, \quad \forall j \neq 1, (k+1), (k+2) \quad (3.6)$$

$$T_i^j \geq T_i^{j-1} + f^{j-1} + s.x_i^{j-1} \quad \forall i, \quad j = (k+1), (k+2) \quad (3.7)$$

$$T_1^1 = 0 \quad (3.8)$$

$$x_i^k + x_i^{k+1} = 1 \quad \forall i \quad (3.9)$$

$$x_i^k, x_i^{k+1} \in \{0, 1\} \quad \forall i \quad (3.10)$$

Amaç fonksiyonunda, daha önce de bahsedildiği gibi, son işin son makinedeki tamamlanma zamanı enküçüklenmek istenmektedir. Yayılma zamanı $k \leq m - 2$ ise formülasyonda da belirtildiği gibi $T_n^m + f^m$ şeklinde; $k = m - 1$ ise $T_n^m + f^m + s.x_n^m$ şeklinde ifade edilir.

(3.2) ve (3.3) bir işin bir makinede işlenmeye başlaması için aynı makinedeki bir önceki parçanın işleminin bitmiş olmasını sağlayan kısıttır. Esnek işlem yapan makinelerde, sabit işlem sürelerinin yanı sıra esnek işlem süreleri de dahil edilmiştir. (3.2) esnek olmayan; (3.3) esnek makineler içindir. (3.4), sıfır ara stoklu sistemlerde bir makinede bir işlemin başlayabilmesi için iki önceki parçanın bir sonraki makinede bitmesi gerektiğini ifade eder. Bu kısıt, bir sonraki makinenin esnek olmadığı durum için geçerlidir. Esnek olması durumunda ise (3.5) bu koşulu sağlamaktadır.

(3.6) ve (3.7), bir parçanın sonraki makinede işleme başlayabilmesi için bir önceki makinede işleminin bitmiş olmasını sağlar. (3.6) esnek olmayan; (3.7) esnek makineler için yazılmıştır. (3.8), ilk işin ilk makinede sıfır anında başladığını gösterir. (3.9), bir işin esnek işleminin sadece bir esnek makineye atanmasını garanti eder.

Oluşturulan eniyileme modeli, Intel core I5 3.2 GHz 4GB 64 bit bir makinede, GAMS CPLEX 12.2 kullanılarak çözdürülmüştür. Başlangıç denemelerinde, özellikle parça sayısı artırıldığında, işlem sayısı arttığı için, çözüm zamanının çok daha fazla arttığı gözlemlenmiştir. 50 parçalı bazı örnekler için ise 24 saat içerisinde optimal çözüme ulaşılamamıştır. Gerçek hayatta, bu tip montaj hatlarında parça sayısının çok daha büyük olduğu düşünüldüğünde problemin çözümü için geliştirilen matematiksel modelin kullanılamayacağı açıktır. Çalışmanın bundan sonraki bölümlerinde hızlı ve kaliteli çözümler üreten çeşitli yöntemler geliştirilecektir.

4. Temel Sonuçlar

Bu bölümde, problem çözümü için geliştirilecek dinamik programlama ve sezgisel çözüm yönteminde kullanılmak üzere, sınırsız veya sıfır ara stok kapasiteli problemler için optimal çözümlerin sahip olduğu çeşitli özellikler anlatılmıştır.

Daha önce Muth [29] ve Pinedo [2] ile tanımlanan ters problem, bu çalışmada ele alınan problem için aşağıdaki şekilde ifade edilebilir:

Ters Problem: Orijinal problem parametreleri f^j, s, n, m ve k olsun. Bu durumda ters problem $\hat{f}^j = f^{m+1-j}, \hat{s} = s, \hat{n} = n, \hat{m} = m, \hat{k} = m - k$ olarak tanımlanır.

Önteorem 1 *Sınırsız veya sıfır ara stok kapasiteli problemler için orijinal ve ters problemin optimal yayılma süreleri birbirine eşittir.*

Orijinal problem için verilen optimal çözüm kullanılarak ters problemin optimal çözümü ya da ters problem için verilen optimal çözüm kullanılarak orijinal problemin çözümü kolaylıkla bulunabilir. Sınırsız veya sıfır ara stok kapasiteli problemler için Pinedo [2]'de de belirtilen ve ilk kez 1979 yılında Muth [29] tarafından ispatlanan önteorem, bu çalışma için de geçerlidir. İspatın detaylı bir şekilde açıklaması Muth [29]'da yer almaktadır ve burada tekrar edilmemiştir. Ancak, ters problemdeki üretim sistemi orijinalin aynadaki görüntüsü olarak düşünüldüğünde önteorem kolaylıkla anlaşılabilir.

Orijinal problem için verilen optimal bir çözümden ters problemin optimal çözümünün bulunma şeklini aşağıdaki gibi göstermek mümkündür:

Orijinal problemin çözümü $x_1^{k*}, x_2^{k*}, \dots, x_n^{k*}$ ise ters problem çözümü $\hat{x}_1^{\hat{k}*} = x_n^{k*}, \hat{x}_2^{\hat{k}*} = x_{n-1}^{k*}, \dots, \hat{x}_n^{\hat{k}*} = x_1^{k*}$ olarak bulunabilir.

Orijinal ve ters problem ilişkisi için aşağıdaki örnek açıklayıcı olacaktır:

Örnek 3 $n = 10, m = 5, k = 2, s = 11, f^1 = 13, f^2 = 15, f^3 = 11, f^4 = 23, f^5 = 17$ olan sınırsız ara stok kapasitesine sahip bir örnek alalım. Bu örnek için optimal çözümde yayılma zamanı 297 ve optimal çizelgede 1, 3 ve 5 parçalarının esnek işlemleri üçüncü; diğer parçaların esnek işlemleri de ikinci makineye atanmıştır.

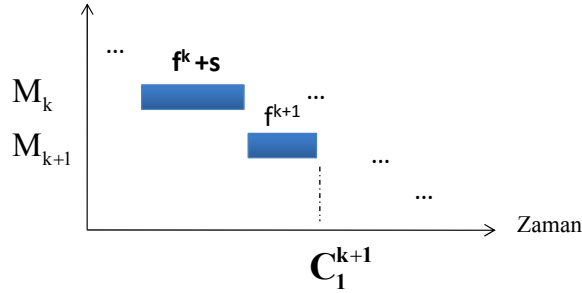
Önteorem 1'den hareketle ters problem parametreleri $\hat{f}^1 = f^5 = 17, \hat{f}^2 = f^4 = 23, \hat{f}^3 = f^3 = 11, \hat{f}^4 = f^2 = 15, \hat{f}^5 = f^1 = 13, \hat{k} = m - k = 3, \hat{s} = s = 11, \hat{n} = n = 10, \hat{m} = m = 5$ şeklindedir. Ters problemin optimal yayılma zamanı 297, atamalar ise 1, 3 ve 5'inci parçaların esnek işlemleri üçüncü, diğer parçaların esnek işlemleri ise dördüncü makineye atanacak şekildedir.

Bu önteorem sayesinde orijinal problemin optimal çözümü biliniyorsa, ters problem için tekrar çözüm bulmaya gerek olmadığı; bu iki problemin optimal çözümlerinin amaç fonksiyonu değerinin aynı olduğu sonucuna ulaşılır. Ters problem özelliğinin çeşitli faydaları vardır. Örneğin, bölüm 6'da problem çözümü için geliştirilen sezgisel çözüm yönteminin etkinliğinin analizi için yapılan deneysel çalışmanın boyutunun azaltılmasını sağlamıştır. Bu deneysel çalışmada belirlenen parametreler dahilinde toplamda 1280 deneme yapılması uygun görülmüştür. Ancak orijinal ve ters problemlerin optimal sonuçlarının eşit olduğunun gösterildiği bu önteorem, test parametrelerinin değiştirilmesi konusunda kolaylık sağlamış ve yapılması gereken deneme sayısı 640'a düşürülmüştür. Örneğin 15 makineli ve esnek makinelerin 1 ve 2 numaralı makineler olduğu bir durum ile 14 ve 15 numaralı makinelerin esnek makineler olduğu durum ters problem ilkesine uygun olduğu için, aynı problemi ifade etmektedir. Bu çeşit durumların optimal sonuçları, bu önteorem yardımı ile bilindiğinden tekrar deneme yapılmasına gerek yoktur. Ayrıca bölüm 5'te anlatılacak olan dinamik programlama yönteminde, m makineli sistemi 3 veya 4 makineli sisteme indirgeme için bu öntelemden yararlanılmaktadır.

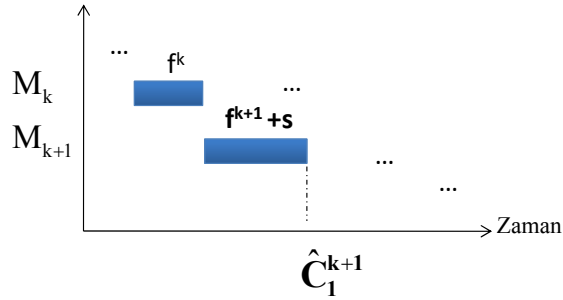
Önteorem 2 *Sınırsız veya sıfır ara stok kapasiteli problemler için optimal çözümde $x_1^k = 0, x_n^k = 1$ 'dir.*

İspat: Sınırsız ara stok kapasiteli bir sistemde, kuralın aksine ilk işin esnek işleminin ilk esnek makineye atandığı optimal bir çizelge olduğunu varsayalım. Bu durumda ilk

işin ilk makinedeki tamamlanma zamanı Şekil 4.1a'da görüldüğü gibi $C_1^k = T_1^k + f^k + s$ ve $C_1^{k+1} = T_1^{k+1} + f^{k+1} = T_1^k + f^k + s + f^{k+1}$ olarak hesaplanır. Şimdi, kurala uygun olacak şekilde, ilk işin esnek işleminin atanmasını ikinci esnek makineye olacak şekilde değiştirelim. Yeni durumda Şekil 4.1b'de de görüldüğü gibi tamamlanma zamanları $\hat{C}_1^k = T_1^k + f^k$ ve $\hat{C}_1^{k+1} = T_1^{k+1} + f^{k+1} + s = T_1^k + f^k + f^{k+1} + s$ olarak bulunur. Esnek makinelere kadar olan makinelerin işlem zamanları sabit olduğu için her iki atama durumunda da $T_1^k = f^1 + f^2 + \dots + f^{k-1}$ olacaktır. $\hat{C}_1^k < C_1^k$ ve $\hat{C}_1^{k+1} = C_1^{k+1}$ olduğu için, ilk yapılan atama optimalse yeni atama optimaldir. Son parçanın esnek işleminin ilk makineye atamasının ispatı Önteorem 1'de ispatlanan ters dönebilirlik özelliğinin doğal bir sonucudur. Bu ispatta kullanılan argümanlar sıfır ara stok kapasiteli problemler için de aynen geçerlidir. \square



a- 1'inci parçanın esnek işlemi ilk makineye atanmış.



b- 1'inci parçanın esnek işlemi ikinci makineye atanmış.

Şekil 4.1: Önteorem 2- İlk parçanın ataması

Bu önteorem geliştirilen dinamik programlama yöntemi ve sezgisel algoritmanın ilk adımı olup, ilk ve son parçaların herhangi bir işlem yapılmadan atamalarının bulunmasını sağlar. Böylece algoritma bu iki parçanın atamasını bulmak için fazladan işlem yapmayacak, alternatiflerin denenmesi nedeni ile yapılacak işlemler çözüm süresinin uzamasına neden olmayacaktır.

Önteorem 3 Sınırsız veya sıfır ara stok kapasiteli problemler için eğer $|f^k - f^{k+1}| \geq s$ ise $x_i^l = 1 \quad \forall i = 2, \dots, n - 1$ ve $l = \operatorname{argmin}_{j \in \{k, k+1\}} \{f^j\}$ 'dir.

İspat: Anuar ve Bukchin [23], $q \in [0, 1]$ için $f^1 + qs = f^2 + (1 - q)s$ (ya da $|f^1 - f^2| \leq s$) koşulu sağlandığında hattın dengelenebilir olduğunu göstermiştir. Önteoremden ise dengelenemeyen hatlarda çözümün kolaylıkla bulunabileceği ifade edilmektedir.

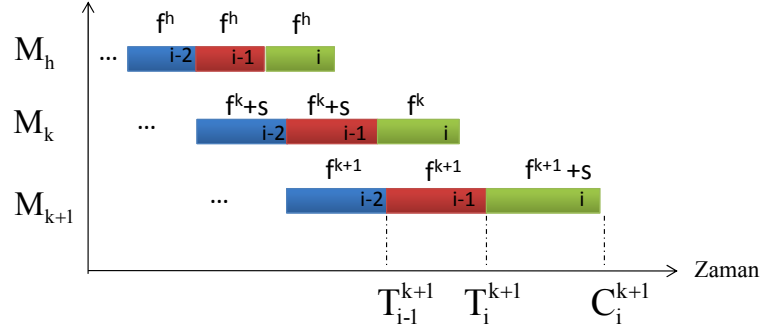
$|f^k - f^{k+1}| \geq s$ durumu $f^k + s \leq f^{k+1}$ ve $f^{k+1} + s \leq f^k$ olarak ayrıştırılabilir. Önteorem 1'deki ters dönelirlik özelliğinden yola çıkılarak bu eşitsizliklerden bir tanesi için önteoremi ispatlamak yeterli olacaktır.

$f^k + s \leq f^{k+1}$ durumunu ele alalım. Bu kuralı ispatlamak için, diğer makinelerin sabit işlem zamanlarının esnek makinelerin sabit işlem zamanları ile kıyaslandığında ortaya çıkabilecek bütün durumlar aşağıda ayrı ayrı ele alınmıştır.

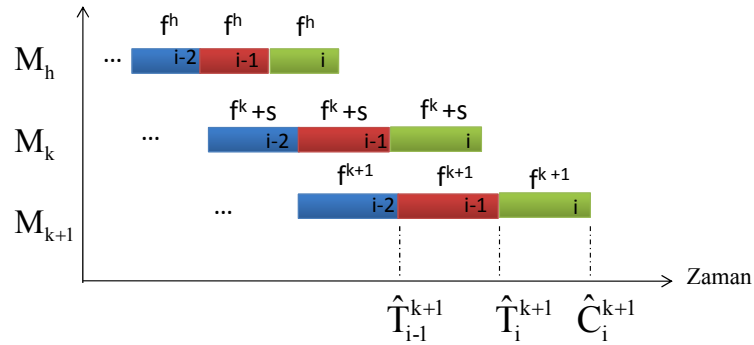
Durum 1: İlk esnek makinenin önünde yer alan bütün makinelerdeki işlem zamanı ilk esnek makinenin sabit işlem zamanına küçük eşit ise, $\forall h = 1, \dots, (k - 1)$
 $f^h \leq f^k$:

Öncelikle, verilen kuralın aksine, en az bir işin, sabit işlem süresi küçük olan makineye atanmamış olduğu optimal bir çözüm olduğunu varsayalım. Bu durumda ilk esnek makinede hiç bir zaman boş zaman oluşmayacaktır. Birden fazla kuralın aksine büyük işlem süreli makineye atanmış esnek işlem olabileceği için i parçası olarak bunlardan en öndekini alalım. İlk işin esnek işlemi Önteorem 2'ye göre ikinci makineye ve son işin esnek işleminin de ilk makineye atanacağı bilindiği için $2 \leq i \leq n - 1$ olsun. Bu durumda oluşan Gantt şeması Şekil 4.2a'da görüldüğü gibi olacaktır. Bu şekilde de görüldüğü gibi h makinesi esnek makinelerden önceki herhangi bir makinedir ve işlem süresi küçük olduğu için sınırsız ara stok durumunda k makinesinde herhangi bir beklemeye neden olmamaktadır. Şimdi, önteoreme uygun şekilde i 'inci işin atamasını ilk esnek makineye kaydıralım. Sadece i 'inci parçanın atamasının değiştiği bu sistemin Gantt şeması Şekil 4.2b'deki gibi olacaktır. $(i - 2)$ ve $(i - 1)$ 'inci parçaların iki esnek makinedeki başlangıç ve bitiş zamanlarında bir değişiklik olmamıştır. $T_i^{k+1} = \hat{T}_i^{k+1}$ 'dir. $f^k + s \leq f^{k+1}$ olduğu için $\hat{C}_i^k \leq \hat{C}_{i-1}^{k+1}$ durumu sağlanacaktır ve bundan sonraki bütün parçalar için de $\hat{C}_l^k \leq \hat{C}_{l-1}^{k+1} \quad \forall l = i + 1, \dots, n$ sağlanacaktır. Bu da

yapılan değişikliğin $k + 1$ makinesindeki işlerin tamamlanma zamanında herhangi bir gecikmeye sebep olmayacağı anlamına gelir.



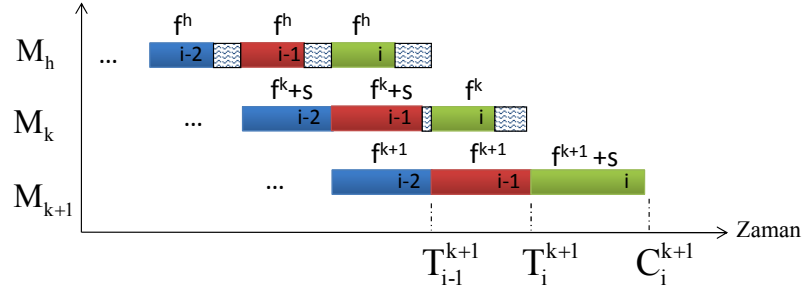
a- i parçasının esnek işlemi ikinci makineye atanmış.



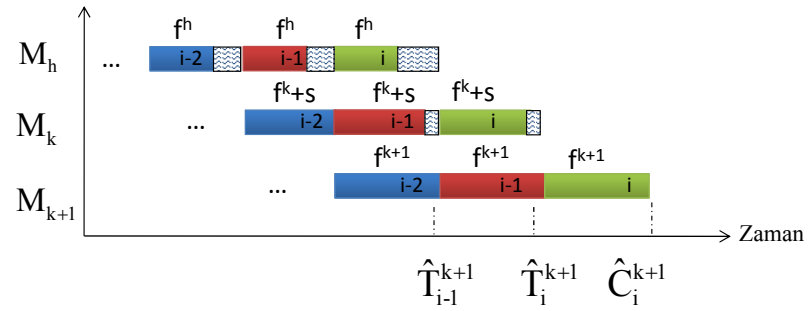
b- i parçasının esnek işlemi ilk makineye atanmış.

Şekil 4.2: Önteorem 3- Durum 1, sınırsız ara stok

Sıfır ara stok problemi için ilk durumun ispatı benzer şekilde olup, Gantt şeması Şekil 4.3'teki gibi olmaktadır. Bu durumda $f^k \geq f^h$ olduğu için h makinesinde boş zaman oluşur. Ancak bu boş zaman k makinesinde parçaların başlangıç ve tamamlanma zamanlarını etkilemez. Sınırsız ara stok probleminde ifade edilen yaklaşımla aynı şekilde, i parçası kuralın aksine işlem süresi büyük olan makineye atanmış ilk parça ve bu çizelge optimal olsun. Şekil 4.3a'da kuralın aksine atama yapılmış olan i parçasının durumu; Şekil 4.3b'de kurala uygun atanmış i parçası görülmektedir. i 'den önceki parçaların başlangıç ve tamamlanma zamanlarında bir farklılık oluşmamış, $T_i^{k+1} = \hat{T}_i^{k+1}$ eşitliği bozulmamıştır. $C_i^{k+1} = T_i^{k+1} + f^{k+1} + s$ ve $\hat{C}_i^{k+1} = \hat{T}_i^{k+1} + f^{k+1}$ olduğu için, Önteorem 3'e uygun atamanın yayılma zamanını artırmayacağı görülür ve ilk durumun ispatı tamamlanmış olur.



a- i parçasının esnek işlemi ikinci makineye atanmış.



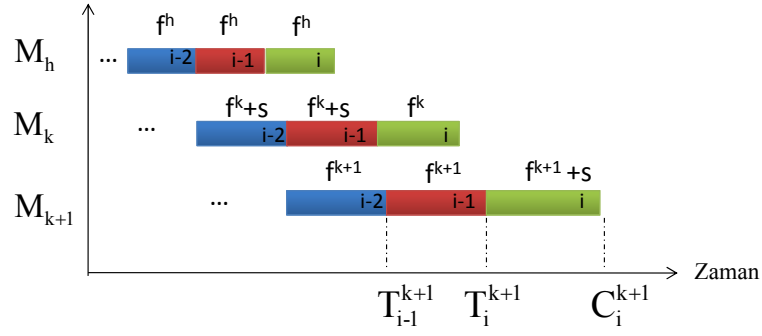
b- i parçasının esnek işlemi ilk makineye atanmış.

Şekil 4.3: Önteorem 3- Durum 1, sıfır ara stok

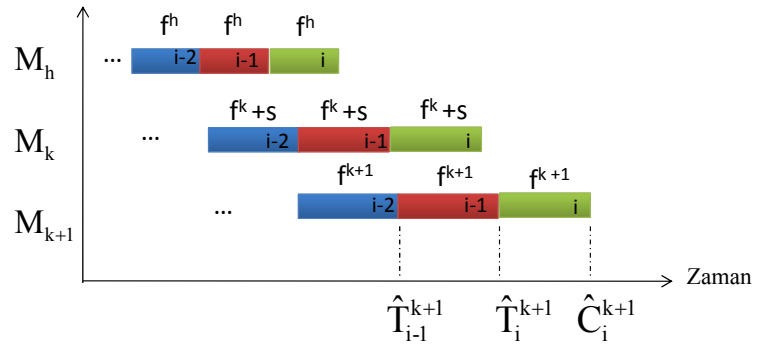
Durum 2: İlk esnek makinenin önündeki makinelerden en az birinin sabit işlem zamanı, ilk esnek makineden büyük; ilk esnek makinenin sabit işlem zamanı ile esnek işlem zamanının toplamına küçük eşit ise, $\exists h \in [1, (k-1)]$

$$f^k < f^h \leq f^k + s:$$

Önce kuralın aksine i 'inci parçanın ataması işlem zamanı büyük olan makineye yapılsın ve i parçası diğer makineye ataması yapılan ilk parça olsun ($2 \leq i \leq n-1$). Bu atamanın optimal olduğunu varsayalım. En az bir makine için $f^k < f^h$ olduğundan bu durum altında ilk esnek makinede boş zaman oluşabilir. Ancak bu boş zaman, $f^h \leq f^k + s$ eşitsizliğinden ve i parçası sıradaki ilk parça olduğu için s 'ten büyük olamaz. Bu durumda Gantt şeması Şekil 4.4a'da görüldüğü gibi olur. Sonra kurala uygun biçimde i parçasının atamasını diğer makineye aldığımızda oluşacak Gantt şeması da Şekil 4.4b'deki gibi olmaktadır. Bu durumda $C_{i-1}^{k+1} > C_i^k$ olur ve $f^k + s \leq f^{k+1}$ varsayımı altında $\hat{T}_i^{k+1} \leq T_i^{k+1} + s$ 'dir. $C_i^{k+1} = T_i^{k+1} + f^{k+1} + s$ ve $\hat{C}_i^{k+1} = \hat{T}_i^{k+1} + f^{k+1}$ olduğundan $C_i^{k+1} \geq \hat{C}_i^{k+1}$ elde edilir. Kurala uygun atamanın ikinci esnek makinedeki tamamlanma zamanını artırmayacağı görülür.

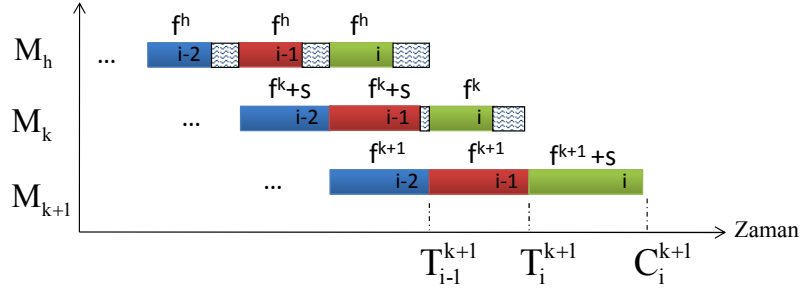


a- i parçasının esnek işlemleri ikinci makineye atanmış.

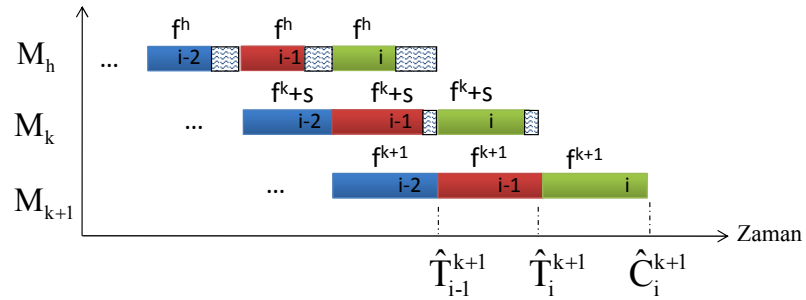


b- i parçasının esnek işlemleri ilk makineye atanmış.

Şekil 4.4: Önteorem 3- Durum 2, sınırsız ara stok



a- i parçasının esnek işlemleri ikinci makineye atanmış.



b- i parçasının esnek işlemleri ilk makineye atanmış.

Şekil 4.5: Önteorem 3- Durum 2, sıfır ara stok

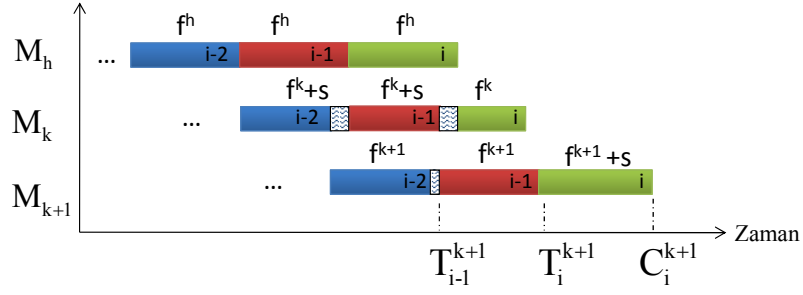
Durum 2 için sıfır ara stok problemi benzer şekilde olup, Şekil 4.5’de görüldüğü gibidir. Buna göre, sınırsız ara stok kapasiteli durumda olduğu gibi $f^k + s < f^{k+1}$, e uygun bir sistem alınsın ve i parçası kuralın aksine esnek işlem süresi büyük olan makineye atanmış ilk parça olsun. Bu durumda karşılaşılabilecek Gantt şeması Şekil 4.5a’daki gibi olur. Ayrıca $f^h > f^k$ olduğu için atamalara bağlı olmak üzere k ’inci makinede boş zaman oluşabilir. Şimdi, i parçasının esnek işlemi Şekil 4.5b’de görüldüğü gibi esnek süresi küçük olan makineye atansın. $f^{k+1} \geq f^k + s$ olduğu için $\hat{T}_i^{k+1} = T_i^{k+1}$ olur. $C_i^{k+1} = T_i^{k+1} + f^{k+1} + s$ ve $\hat{C}_i^{k+1} = \hat{T}_i^{k+1} + f^{k+1}$ olduğu için $\hat{C}_i^{k+1} < C_i^{k+1}$ elde edilir. Kurala uygun atama, bu ve bundan sonraki hiç bir parça için esnek makinelerin ikincisindeki tamamlanma zamanını artırmayacağı için, ikinci durumun ispatı da tamamlanmış olur.

Durum 3: İlk esnek makineden önceki makinelerden en az birinin sabit işlem zamanı ilk esnek makinenin sabit işlem zamanı ve esnek işlem zamanı toplamından büyük ise, $\exists h \in [1, (k - 1)] f^k + s < f^h$:

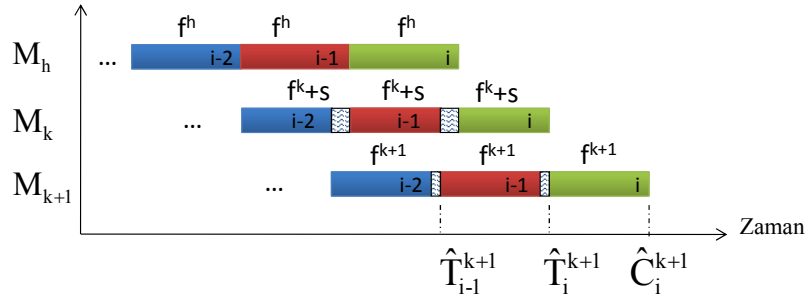
Kuralın aksine esnek işlemi ikinci makineye atanan en az bir parçanın olduğu optimal bir çizelge olsun. Bu parçalardan en önde olan i parçasını ele alalım ($2 \leq i \leq n - 1$). $f^h > f^k + s$ olduğu için ikinci esnek makinede boş zaman oluşacaktır. İkinci esnek makinenin başlangıç ve bitiş zamanları f^h ile f^{k+1} arasındaki ilişkiye göre değişir. Bu ilişkiyi, sınırsız ara stok kapasitesine sahip sistemler için $f^h \geq f^{k+1}$ ve $f^h < f^{k+1}$ olarak iki durum halinde inceleyecek olursak;

Durum 3.1: İlk esnek makineden önceki makinelerden en az birinin sabit işlem zamanı ikinci esnek makinenin sabit işlem zamanından büyük ise, $\exists h \in [1, (k - 1)] f^h > f^{k+1}$:

Bu durumda ikinci esnek makinede de boş zaman oluşur. Elde edilecek Gantt şeması Şekil 4.6a’daki gibi olmaktadır. Şimdi, i parçasının esnek işlemini ilk esnek makineye kaydıralım. Yeni Gantt şeması Şekil 4.6b’deki gibidir. $f^h > f^k + s$ olduğu için $T_l^k = \hat{T}_l^k \forall l$ geçerli olacaktır. Diğer taraftan $f^{k+1} \geq f^k + s$ olduğu için atama değiştikten sonra $i - 1$ ve i parçaları arasında ikinci esnek makinede oluşacak boş zaman en fazla s kadar olabilir. Bu yüzden $\hat{C}_i^{k+1} \leq C_i^{k+1}$ olacaktır.



a- i parçasının esnek işlemi ikinci makineye atanmış.



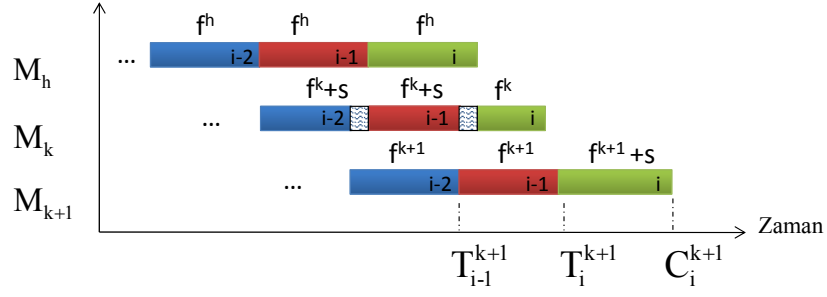
b- i parçasının esnek işlemi ilk makineye atanmış.

Şekil 4.6: Önteorem 3- Durum 3 $f^h > f^{k+1}$, sınırsız ara stok

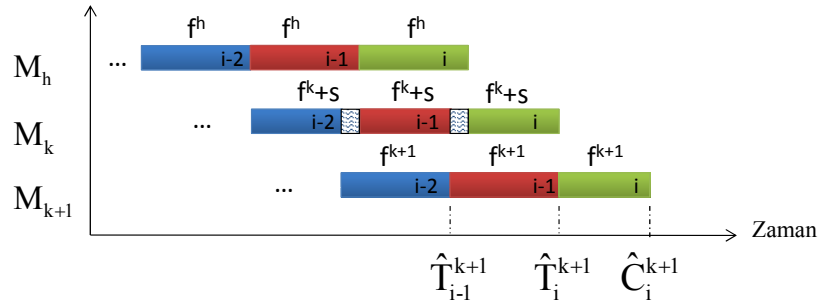
Durum 3.2: İlk esnek makineden önceki makinelerden en az birinin sabit işlem zamanı ikinci esnek makinenin sabit işlem zamanına küçük eşit ise, $\exists h \in [1, (k-1)]$ $f^h \leq f^{k+1}$:

i 'inci parçanın ataması, Şekil 4.7'de görüldüğü gibi, ilk makineye kaydırıldığında $f^h \leq f^{k+1}$ ise ikinci esnek makinede boş zaman oluşmaz ve $\hat{T}_i^{k+1} = T_i^{k+1}$ eşitliği bozulmaz. $f^h > f^k + s$ olduğu için $T_i^k = \hat{T}_i^k$, $\forall l$ hala geçerlidir. $C_i^k > \hat{C}_i^{k+1}$ olduğu için bu atama tamamlanma zamanını artırmaz.

Sıfır ara stok problemi için 3'üncü durum tek bir Gantt şeması ile gösterilebilir. i parçasının esnek işlemi kurala uymayacak şekilde sabit işlem süresi büyük olan makineye atansın ve bu i parçası sistemde bu şekilde bir atamanın yapıldığı ilk parça olsun. Bu atama ile oluşan Gantt şeması Şekil 4.8a'da görülmektedir. Şimdi, kurala uygun bir şekilde i parçasının atamasını diğer makineye alalım. Şekil 4.8b'de görüldüğü gibi $C_i^k \leq C_{i-1}^{k+1}$ 'dir. Dolayısıyla $f^k + s \leq f^{k+1}$ olduğu için i parçasının iki durumda da $k+1$ 'inci makinedeki başlangıç zamanları aynıdır ($T_i^{k+1} = \hat{T}_i^{k+1}$). i parçasının tamamlanma zamanları ise $C_i^{k+1} = T_i^{k+1} + f^{k+1} + s$ ve $\hat{C}_i^{k+1} = \hat{T}_i^{k+1} + f^{k+1}$ olarak hesaplanır. Bu durumda $\hat{C}_i^{k+1} < C_i^{k+1}$ olduğu için, kurala uygun atamanın



a- i parçasının esnek işlemi ikinci makineye atanmış.



b- i parçasının esnek işlemi ilk makineye atanmış.

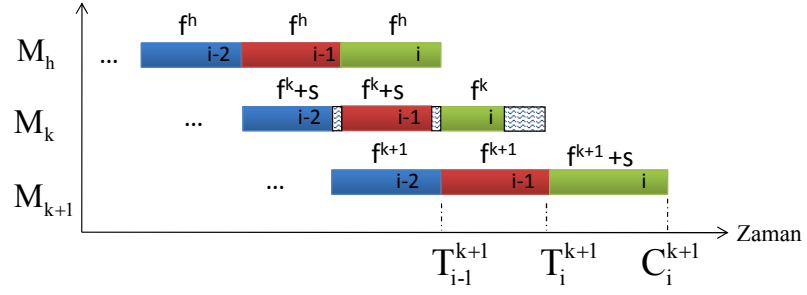
Şekil 4.7: Önteorem 3- Durum 3 $f^h \leq f^{k+1}$, sınırsız ara stok

ikinci esnek makinedeki tamamlanma zamanını artırmayacağı görülür. Böylece ispat tamamlanmış olur. \square

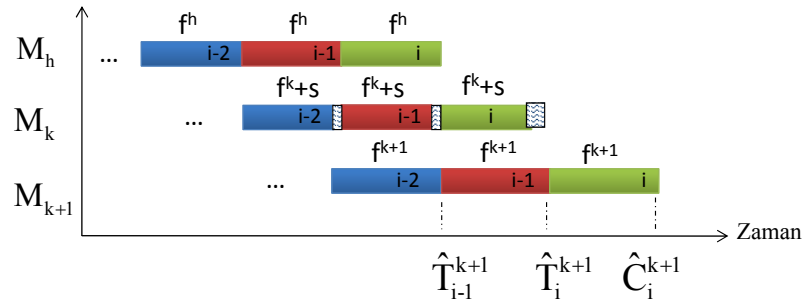
Bu önteorem ile problemin $|f^k - f^{k+1}| \geq s$ eşitsizliğine uyan örnekleri için herhangi bir yöntem kullanılmadan optimal sonuç bulunmuş olur. Buna göre ilk ve son parçaların ataması Önteorem 2'ye göre yapılır. Diğer bütün parçaların esnek işlemleri ise, sabit işlem zamanı daha küçük olan esnek makineye atanır. Bu duruma uyan problemlerin çözümü bellidir. Geliştirilecek çözüm yönteminin çözüm alanının, bu tip örnekleri kapsamamasına gerek yoktur.

$|f^1 - f^2| = s$ olduğunda ise mükemmel denge sağlanmaktadır. Bütün parçaların her iki makinedeki işlem zamanının birbirine eşit olması ve dolayısıyla makinelerde hiç boş zaman oluşmaması durumuna mükemmel denge denir. Bu koşul dahilinde tüm işlerin, esnek işlem süresi küçük olan makineye atanması durumunda mükemmel denge elde edilmiş olur.

Aşağıda ispatlanan diğer iki önemli sonuç belirli şartlar sağlandığında ele alınan problem çözümünün çok kolay olduğunu göstermektedir.



a- i parçasının esnek işlemi ikinci makineye atanmış.



b- i parçasının esnek işlemi ilk makineye atanmış.

Şekil 4.8: Önteorem 3- Durum 3, sıfır ara stok

Önteorem 4 Sınırsız veya sıfır ara stok kapasiteli problemlerde $f^k + s \leq f^l$ ve $f^{k+1} + s \leq f^l$, $\exists l \neq k, k+1$ ise, problem kolaylıkla çözülür.

İspat: Bu durumda esnek işlemlerin hangi makineye atandığının bir önemi yoktur. Yayılma zamanını l makinesi belirler. Yayılma zamanı atamadan bağımsız olarak $C_{max} = f^1 + f^2 + \dots + f^{l-1} + n f^l + f^{l+1} + \dots + f^n + s$ 'dir. Bu değer sınırsız veya sıfır ara stok kapasiteli problemler için değişmemektedir. \square

Bu önteorem ile kurala uyan problem örnekleri için herhangi bir işlem yapılmadan optimal çözüm bulunmuş olur. Çözüm yöntemlerinin çalıştırılıp sonuçların değerlendirilmesine gerek yoktur.

Önteorem 5 Sınırsız veya sıfır ara stok kapasiteli problemler için $n, m, k, f^1, \dots, f^m, s$ verilmiş olsun. Eğer $f^h < f^k \forall h = 1, \dots, k-1$ ise problem $\hat{n} = m - k + 1, \hat{k} = 1, \hat{f}^l = f^{k+l-1}$ ($l = 1, \dots, m - k + 1$), $\hat{s} = s, \hat{n} = n$ şekline; eğer $f^h < f^{k+1}, \forall h = k+2, \dots, m$ ise problem $\hat{n} = k + 1, \hat{k} = k, \hat{f}^l = f^l$ ($l = 1, \dots, k + 1$), $\hat{s} = s, \hat{n} = n$ şekline indirgenebilir.

İspat: Sınırsız veya sıfır ara stok kapasiteli problem için ilk esnek makineden önceki makinelerin sabit işlem zamanları ilk esnek makinenin sabit işlem zamanından küçükse, esnek işlem yapmayan makineler esnek işlemin atanmasını değiştirmeyeceği için yok sayılarak atamalar belirlenir. Elde edilen çözüme bu makinelerdeki işlem süreleri eklenerek optimal yayılma zamanı bulunur. Makine sayısı azaldığı için olası varyasyonlar azalır ve problemin boyutu küçülür. Önteorem 1'deki ters dönebilirlik özelliğinden faydalanılarak ikinci esnek makineden sonraki makinelerin sabit işlem zamanları, ikinci esnek makinenin sabit işlem zamanından küçükse de aynı durum geçerlidir. □

Bu önteoremin sonucu olarak, verilen şartların geçerli olduğu durumlarda, orijinal problemdeki makine sayısı yerine indirgenmiş problem çözülebilir. Bu kurala uyan makineler göz ardı edilerek problem yeniden düzenlenir. Problem boyutu küçüldüğü için algoritmanın hızı artar, daha kısa sürede uygun sonuçlar elde edilebilir. Ayrıca esnek makineler haricindeki tüm makinelerin işlem zamanları esnek makinelerin sabit işlem zamanından küçük olduğu durumda problem 2-makine sistemine dönüşür. Bu problem çözümü için Crama ve Gültekin [24]'in önerdiği kesin sonuç veren çözüm yöntemi kullanılabilir. Makine sayısı azaltmayı sağlayan bu önteorem özellikle makine sayısının çok fazla olduğu problemlerin çözüm süresini azalttığı için oldukça önem taşımaktadır.

Bir sonraki bölümde önteoremlerin belirttiği optimal çözüm niteliklerinden yararlanılarak geliştirilmiş, yapay polinom zamanlı kesin çözüm veren dinamik programlama yaklaşımı anlatılacaktır.

5. Dinamik Programlama

Bu bölümde, m makineli esnek akış tipi üretim sistemlerinin sınırsız veya sıfır ara stok kapasiteli durumları için optimal çözümü veren dinamik programlama algoritmaları geliştirilmiştir. Sınırsız veya sıfır ara stok problemleri için, öncelikle 3 makineli ve 4 makineli sistemlerin olası durumları incelenmiş ve m makineli genel problemin 3 veya 4 makineli probleme indirgenip bu problemler için geliştirilen yöntemlerle çözülebileceği gösterilmiştir. Geliştirilen dinamik programlama yöntemleri bir önceki bölümde ispatlanmış olan temel sonuçları kullanmaktadır.

5.1 Dinamik Programlama Genel Durum

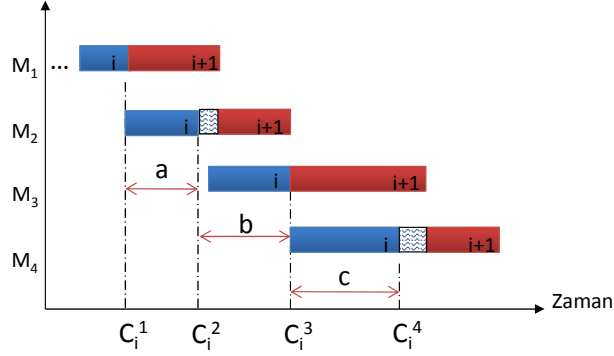
Önteorem 3'te $|f^k - f^{k+1}| \geq s$ durumunun çözümünün kolay olduğu ispatlanmıştır. Dolayısıyla bu bölüm boyunca $|f^k - f^{k+1}| < s$ durumu ele alınacaktır. Ayrıca Önteorem 4'te ispatlanan $\exists l \neq k, k+1$ $f^k + s \leq f^l$ ve $f^{k+1} + s \leq f^l$ durumları dışında kalan problemler ele alınacaktır. Çünkü bu durumlara uygun problemlerin çözümleri bilinmektedir. Ayrıca işlem zamanlarının (f^j, s) tamsayı olduğu varsayılmaktadır. Bu varsayım, çizelgeleme literatüründe sıklıkla kullanılan bir varsayımdır. Gerçekte tamsayı olmayan değerler uygun katsayılarla çarpılarak tamsayıya çevrilebilir.

Dinamik programlama süresince kullanılacak notasyonlar aşağıdaki gibidir:

- S_i : i 'inci aşamadaki durum ($i = 1, \dots, n$). S_i durumu 3 makineli sistemler için (a, b) şeklinde ikili; 4 makineli sistemler için (a, b, c) şeklinde üçlü değerler olarak ifade edilecektir.

a, b ve c değerleri herhangi bir işin (i 'inci iş) ardışık iki makine arasındaki (j 'inci ve $j+1$ 'inci makine) tamamlanma zamanı farklarını göstermektedir. 3 makineli

sistemde a değeri 1 ve 2; b değeri 2 ve 3 numaralı makineler arasındaki zaman farklarını ifade ederken, 4 makineli sistemlerde bunlara ek olarak 3 ve 4 numaralı makineler arasındaki zaman farkını ifade eden c değeri de bulunmaktadır ($a = C_i^2 - C_i^1$, $b = C_i^3 - C_i^2$ ve $c = C_i^4 - C_i^3$). a, b ve c değerleri Şekil 5.1’de daha net bir şekilde görülmektedir.



Şekil 5.1: 4 makineli sistemde (a,b,c) değerleri

Bu şekilde tanımlanan durum bilgisi kullanılarak dinamik programlama algoritmasının temelini oluşturan optimallik şartı şu şekilde ifade edilebilir:

“Optimal bir çizelgede, i aşamasında (i parçası için) S_i durumu ortaya çıkmışsa bu çizelge, bu aşamada ortaya çıkabilecek durumlar içinde $C_n^m - C_i^m$ farkı en küçük olandır.”

Bu ifadeye göre, bir optimal çizelgede bulunan i parçasından sonraki optimal çizelgeyi bulabilmek için gerekli tüm bilgi a, b, c değerlerinde gizlenmiştir. Çizelgenin bu parçadan önceki kısmının bilinmesine gerek yoktur.

Bunun yanında, ilerleyen bölümlerde de görüleceği gibi, herhangi bir i aşamasında ortaya çıkabilecek S_i durumlarının sayısı, problem parametrelerinin tamsayı olması halinde sonlu sayıdadır ve bu geliştirilecek algoritmanın yapay polinom zamanda çözüm üretmesini sağlamaktadır.

a, b ve c değerleri problem verilerine göre çok farklı değerler alabilmektedir. Bu nedenle, her bir parça için ayrı ayrı olabilecek tüm değerleri içeren aralıklar belirlenmiştir. Bu aralıklar, makine sayısı ve esnek makinelerin sistemdeki konumuna göre farklı şekillerde hesaplanacaktır.

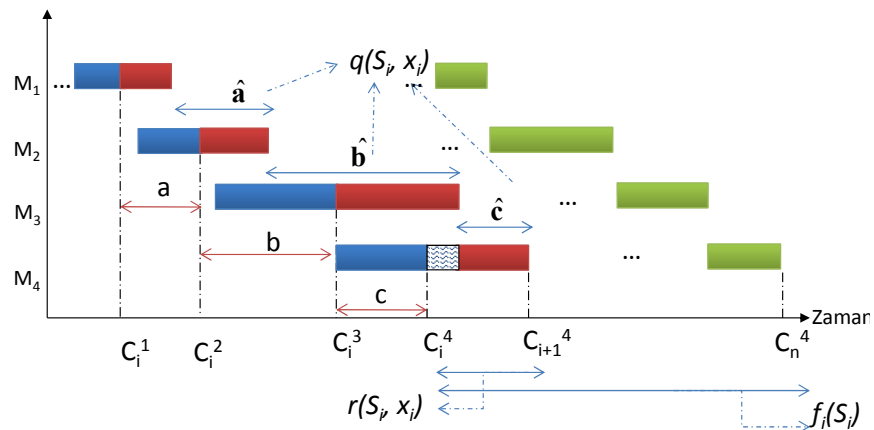
- x_i : i 'inci aşamadaki karar değişkeni. x_i (0-1) değerlerini alan bir değişkendir. 0 olması durumu esnek işlemin ilk esnek makineye atanmış; 1 olması durumu da ikinci esnek makineye atanmış olması anlamına gelmektedir.

- $r(S_i, x_i)$: i 'inci aşamada S_i durumu varken x_i karar değişkenine göre i ile $i + 1$ parçasının tamamlanma zamanları arasındaki fark ($C_{i+1}^m - C_i^m$).
- $f_i(S_i)$: i 'inci aşama S_i durumunda i parçası ile n parçasının tamamlanma zamanları arasındaki minimum fark. ($C_n^m - C_i^m$).
- $q(S_i, x_i)$: i 'inci aşama S_i durumunda ise x_i kararı verildiğinde S_{i+1} 'in elde edeceği değer.

Bu bilgiler ışığında, i 'inci aşamadaki bu parçadan sonraki parçaları içeren minimum tamamlanma zamanı yenileme denklemi olarak aşağıdaki gibi hesaplanabilir:

$$f_i(S_i) = \min_{x_i \in \{0,1\}} \{r(S_i, x_i) + f_{i+1}(q(S_i, x_i))\}$$

Önteorem 2'ye göre başlangıç durumu $x_n = 0$ 'dır. n 'inci parçanın ataması için $f_n(S_n)$ değeri olabilecek bütün $S_n = (a, b)$ değerlerine $x_n = 0$ uygulanarak bulunur. Yani burada, dinamik program son parçadan başlayıp ilk parçaya doğru, geri yönde işlemektedir. Böylece i 'inci parçadan n 'inci parçaya kadar olan minimum toplam tamamlanma zamanı, i 'inci parçanın tamamlanma zamanı ile işlem yapılmış bir önceki aşama olan $i + 1$ 'inci parçadan n 'inci parçaya kadar elde edilen, minimum toplam tamamlanma zamanların toplanması ile bulunmaktadır. 4 makineli bir sistem için notasyonların gösterimi Şekil 5.2'deki gibidir.



Şekil 5.2: 4 makineli sistemde notasyonların gösterimi

Toplam tamamlanma zamanları, her bir aşamada bir önceki aşama için hesaplanmış tamamlanma zamanları üzerine konularak elde edilir. Tüm sistemin yayılma zamanı ancak ilk parçaya gelindiğinde hesaplanabilir.

Önteorem 2'ye göre ilk parçanın esnek işleme ikinci esnek makineye ($k + 1$); son parçanın esnek işleme ilk esnek makineye (k) atanır. Bu sebeple $x_n = 1$ ve $x_1 = 0$ 'dır.

3, 4 ve m makineli sistemlerde (a, b, c) 'nin alabileceği bütün değerlerin ve tamamlanma zamanlarının hesaplanması farklılık göstermekle birlikte dinamik programın işleyişi genel olarak aşağıdaki gibi olmaktadır:

- Problem parametrelerine göre a, b ve c 'nin alabileceği bütün değerler belirlenir.
- Son parçanın ataması ilk esnek makineye yapılır (Önteorem 2).
- Her bir parçanın atamasını bulmak için, son parçadan geriye doğru gidilir. Her bir durum için o parçadan sonraki minimum toplam tamamlanma zamanları belirlenir. Her bir aşamadaki hesaplama sadece, bir önceki aşamada yapılan hesaplamalar kullanılır. İlk parçaya gelene kadar aynı şekilde devam edilir.
- İlk parçanın ataması ikinci esnek makineye yapılarak (Önteorem 2) optimal yayılma zamanı hesaplanır.

Sondan başa doğru ilerleyen bu çözüm yönteminde tamamlanma zamanları arasındaki farklar kümülatif olarak hesaplanır. n 'inci parçadan 2'nci parçaya kadar tamamlanma zamanları ve ulaşılabilecek durumlar belirlenir. İlk parçanın ataması Önteorem 2'de belirtildiği gibi ikinci esnek makineye yapılır. Ataması bilindiği için kolaylıkla hesaplanan ilk işin son makinedeki tamamlanma zamanına ($C_1^m = f^1 + f^2 + \dots + f^m + s$) hesaplanan değerler eklendiğinde, optimal çözümün yayılma zamanı bulunmuş olur.

Geliştirilen dinamik programlama yönteminin çözüm süresi, a, b ve c değerlerinin alabileceği değişik değerlerin sayısı ve üretilecek toplam parça sayısı ile değişmektedir. a, b, c 'nin alabileceği değerlerin sayısı ise sabit ve esnek işlem süreleri (f^j) ve (s) ile parça sayısına (n) bağlıdır. Yapay polinom zamanlı bu çözüm yöntemi, özellikle işlem sürelerinin küçük olduğu durumlarda kısa sürede çözüm vermektedir. Bu bölümün geri kalan kısmında, sınırsız veya sıfır ara stok problemleri için, m makineli genel problemi çözmekte kullanılacak 3 ve 4 makineli sistemleri ve özel durumları tek tek ele alınacaktır.

5.2 Sınırsız Ara Stok Problemi

m makineli sistemin çözümü için birer geçiş basamağı olan 3 ve 4 makineli sistemler, esnek makinelerin sistemdeki konumlarına göre çeşitli özel durumlar oluştururlar. 3 makineli esnek akış tipi üretim sistemi, 2 esnek makine ve bir esnek olmayan makineden oluşan sistem olarak tanımlanabilir. Esnek makinelerin ardışık olma koşulu varsayımlar arasında olduğu için, esnek makineler 1'inci ve 2'nci makineler veya 2'nci ve 3'üncü makineler olabilir. 4 makineli sistemde ise esnek makineler 2'nci ve 3'üncü makineler olarak ele alınacaktır. Çünkü 4 makineli sistemlerde esnek makinelerin 1'inci ve 2'nci ile 3'üncü ve 4'üncü makineler olduğu durumlar, ilerleyen bölümlerde anlatılacak yöntemlerle m makineli sistemlerde olduğu gibi, 3 makineli sisteme indirgenebilmektedir. Sonraki bölümlerde, bu durumların ayrı ayrı incelenmesi ile oluşturulmuş dinamik programlama yönteminin yenileme denklemleri ve durum geçişleri, çözüm yöntemleri anlatılmaktadır.

5.2.1 3 Makineli Sistem, Esnek Makinelerin 1 ve 2 Olduğu Durum ($m = 3, k = 1$)

$S_i = (a, b)$ durumunda $C_i^2 - C_i^1$ olarak tanımlanmış olan a 'nın alabileceği en küçük değer ikinci makinenin sabit işlem zamanına eşittir (f^2). Alabileceği en büyük değer için bir üst sınır ise şu şekilde bulunabilir: Bütün parçaların esnek işleminin ikinci makineye atandığı varsayılırsa, Önteorem 3'ün dışında kalan durumlar ele alındığından, $f^2 + s > f^1$ olduğu için n parçasına gelindiğinde ikinci ve birinci makinelerdeki tamamlanma zamanları arasındaki fark maksimum olacaktır. Dolayısıyla bu değer $n(f^2 + s - f^1) + f^1$ olacaktır. a hiç bir zaman bu değerden büyük bir değer alamaz. Sonuç olarak a 'nın alabileceği değerler $[f^2, n(f^2 + s - f^1) + f^1]$ aralığındaki bütün tamsayı değerlerini içermektedir. Benzer şekilde $C_i^3 - C_i^2$ olarak tanımlanmış olan b 'nin alabileceği en küçük değer üçüncü makinedeki sabit işlem süresine (f^3) eşittir. b 'nin alabileceği en büyük değer ise şu şekilde bulunabilir: n parçasının üçüncü ve ikinci makineler arasındaki maksimum tamamlanma zamanı farkı Önteorem 3'ten yararlanılarak tüm parçaların ilk makineye atanması ile bulunur. Bu nedenle, bu değer $n(f^3 - f^2) + f^2 + s$ ile bulunur. Böylece b 'nin alabileceği değerler $[f^3, n(f^3 - f^2) + f^2 + s]$ aralığındaki bütün tamsayılardır.

i 'inci aşama $S_i = (a, b)$ durumundaki tamamlanma zamanları farkı aşağıdaki formüllerle hesaplanabilir:

- Esnek işlemin ilk makineye atandığı durumda ($x_i = 0$),

$$r((a, b), 0) = \max\{0, \max\{0, f^1 + s - a\} + f^2 - b\} + f^3$$

Bu formüle göre i parçasının esnek işlem ataması ilk makineye yapılmıştır. Eğer a değeri ilk makinedeki işlem süresinden büyükse; parçanın ilk makinede işlemi tamamlandıktan sonra arada bekleme olmadan ikinci makineye geçer. İkinci makinedeki bitiş süresi, bu süreye işlem süresi eklenerek bulunur. Eğer a değeri ilk makinedeki işlem süresinden küçükse, ikinci makinede $f^1 + s - a$ kadar bir bekleme olur, ardından ikinci makinedeki işlem süresi eklenerek ikinci makinedeki tamamlanma zamanı (C_i^2) hesaplanır. Aynı durum parçanın ikinci makineden üçüncü makineye geçişinde de geçerlidir.

- Esnek işlemin ikinci makineye atandığı durumda ($x_i = 1$),

$$r((a, b), 1) = \max\{0, \max\{0, f^1 - a\} + f^2 + s - b\} + f^3$$

Bu formülde de yukarıda anlatılan durumlar geçerlidir. Tek fark, atamaya göre esnek işlem süresinin eklendiği makinenin değişmesidir.

Geriye doğru çalışan bu çözüm yönteminde, i 'inci aşamaya hangi durumdan ulaşılabileceğini bulmak için, $i + 1$ 'inci aşamadaki durumun bilinmesi gerekir. $i + 1$ 'inci aşamadaki durumu bulmak için ise aşağıda belirtilmiş formüller kullanılabilir. Bu formüller, i parçasının ilk ve ikinci makineye atanmış olması durumlarını ele alır. İlgili durum için, ilk makineden başlayarak her bir makinedeki olası bütün a ve b 'leri değerlendirir.

Bulunan durum ve verilen karara göre ulaşılabilecek yeni durumu ifade eden $S_{i+1} = (\hat{a}, \hat{b}) = q((a, b), x_i)$ şu şekilde hesaplanır:

- $S_i = (a, b)$ durumunda esnek işlem ilk makineye atanmış ise ($x_i = 0$)

$$q((a, b), 0) = (\hat{a}, \hat{b}) = (\max\{f^2, a - (f^1 + s) + f^2\}, \max\{f^3, b - \max\{f^2, a - (f^1 + s) + f^2\} + f^3\})$$

$i + 1$ 'den i 'ye geçişte kullanılan bu formül, atama çizelgesinin belirlenmesini sağlar. Formülün ilk terimi ile \hat{a} elde edilir. \hat{a} 'nın alabileceği değer, ikinci

makinede boş zaman yoksa ikinci makinenin sabit işlem süresine eşit olabilir (f^2). Eğer \hat{a} değeri ilk makinedeki işlem zamanından büyükse, ikinci makinede boş zaman oluşabilir ($a - (f^1 + s)$ kadar), bu zamana ikinci makinenin işlem süresi eklenerek olabilecek değer elde edilir. Formülün ikinci terimi ile elde edilecek \hat{b} değeri için, ilk terimde hesaplanan \hat{a} değerinin üzerine eklemeler yapılarak 3'üncü makinedeki durumlar bulunur. Üçüncü makinede boş zaman yoksa f^3 doğrudan eklenir. Boş zaman oluşmuşsa ($\hat{b} - \hat{a}$) ikinci makinedeki zamana f^3 eklenerek \hat{b} değeri hesaplanır.

- $S_i = (a, b)$ durumunda esnek işlem ikinci makineye atanmış ise ($x_i = 1$)
 $q((a, b), 1) = (\hat{a}, \hat{b}) = (\max\{f^2 + s, a - f^1 + f^2 + s\}, \max\{f^3, b - \max\{f^2 + s, a - f^1 + f^2 + s\} + f^3\})$

Bu formülün de diğerinden farklı, atamaya göre değişen esnek işlem süresinin eklendiği makinedir.

Bu iki alternatif $\hat{a} = \max\{f^2 + s.x_i, a - (f^1 + s.(1 - x_i)) + f^2 + s.x_i\}$ ve $\hat{b} = \max\{f^3, b - \max\{f^2 + s.x_i, a - (f^1 + s.(1 - x_i)) + f^2 + s.x_i\} + f^3\}$ şeklinde özetlenebilir.

$m = 3, k = 1$ problemi için yenileme denklemi aşağıdaki gibidir:

$$f_i(a, b) = \min_{x_i \in \{0,1\}} \{(r(a, b), x_i) + f_{i+1}(q((a, b), x_i))\}$$

Her bir aşamada olabilecek her durumda, o parçanın tamamlanma zamanı ve o parçadan sonraki bütün parçaların toplam tamamlanma zamanları hesaplanır. İlk parçanın ataması Önteorem 2'de olduğu gibi ikinci makineye yapılacağı için ilk parçada herhangi bir durum incelemesi yapılmasına gerek yoktur, ilk parçada karşılaşılabilecek tek durum $(a, b) = (f^2 + s, f^3)$ olacaktır. Daha sonra aşamalardaki en küçük tamamlanma zamanlarını veren durumlar belirlenir. Bu durumlardaki atamalar problemin optimal çizelgesini oluştururken, geriye doğru gidilerek ilk parçada hesaplanan minimum tamamlanma zamanı, problemin yayılma zamanını vermektedir. $m = 3, k = 1$ problemi için geliştirilmiş bu dinamik yöntemin çözüm süresi $O(n^3(f^2 - f^1)(f^3 - f^2) + n^2(f^2 - f^1) + n^2(f^3 - f^2))$ 'dir. Bu çözüm süresi, a ve b değerlerinin içinde olduğu aralıkların boyutu ile $n - 1$ 'in çarpılması sonucu elde edilmiştir.

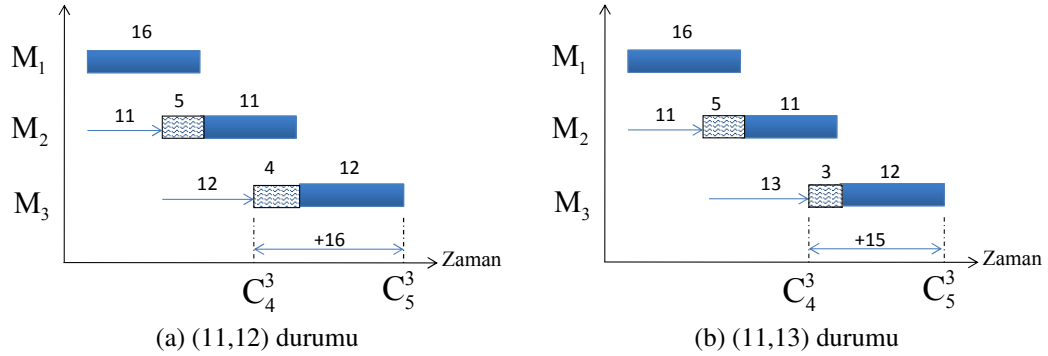
Örnek 4 Dinamik programlama yönteminin işleyişini daha iyi görebilmek için $n = 5, m = 3, k = 1, f^1 = 9, f^2 = 11, f^3 = 12, s = 7$ olan bir örnek alalım. a ve b 'nin değerlerini bulabilmek için veriler $m = 3, k = 1$ için yazılmış formüllerde yerine konulur. a 'in alabileceği değerlerin $[11,54]$ aralığındaki; b 'nin alabileceği değerlerin ise $[12,21]$ aralığındaki tamsayı değerleri olduğu görülür. Bu bilgiler daha anlaşılabilir olması için bir Çizelge 5.1'de gösterilmiştir. Çizelgenin satırlarını (a, b) 'ler oluştururken, sütunlarında her bir aşama (parça) yer alır.

	1	2	3	4	5
(11,12)	x	-	-	-	-
(11,13)	x	-	-	-	-
(11,14)	x	-	-	-	-
...	...	-	-	-	-
(11,21)	x	-	-	-	-
(12,12)	x	-	-	-	-
...	...	-	-	-	-
(18,12)	-	-	-	-	-
...	...	-	-	-	-
(54,12)	x	-	-	-	-
(54,13)	x	-	-	-	-
...	...	-	-	-	-
(54,21)	x	-	-	-	-

Çizelge 5.1: Dinamik programlama Örnek 4-durumlar ve aşamalar

Dinamik programlama yöntemi son parçadan başlayarak ilk parçaya doğru ilerlediği için, çizelge 5'inci aşamadan (parçadan) doldurulmaya başlanır. Öntem 2'ye göre son parçanın esnek işleminin atamasının ikinci makineye, ilk parçanın esnek işlem atamasının ise ilk makineye yapılacağı bilinmektedir. Bu yüzden değerlendirilecek durumlar 5'inci parçanın sadece ilk makineye atanması durumu ile başlayıp ilk parçanın ikinci makineye atanması ile tamamlanmalıdır.

5'inci parça ilk makineye atandığında işlem süreleri sırasıyla 16, 11 ve 12 olacaktır. İlk durum olan (11,12) ile bu parçadan elde edilecek tamamlanma zamanı $r((a, b), 0) = \max\{0, \max\{0, f^1 + s - a\} + f^2 - b\} + f^3$ formülü ile hesaplanabilir. a ve b değerleri olan 11 ve 12 yerine konulduğunda elde edilen tamamlanma zamanı farkı Şekil 5.3a'da görüldüğü gibi 16 olarak bulunmaktadır. Sonraki durum olan (11,13) durumundaki tamamlanma zamanları farkı ise Şekil 5.3b'de görüldüğü gibi, yine aynı formülde a ve b yerine konularak, 15 olarak bulunur. 5'inci parça için tüm durumlar

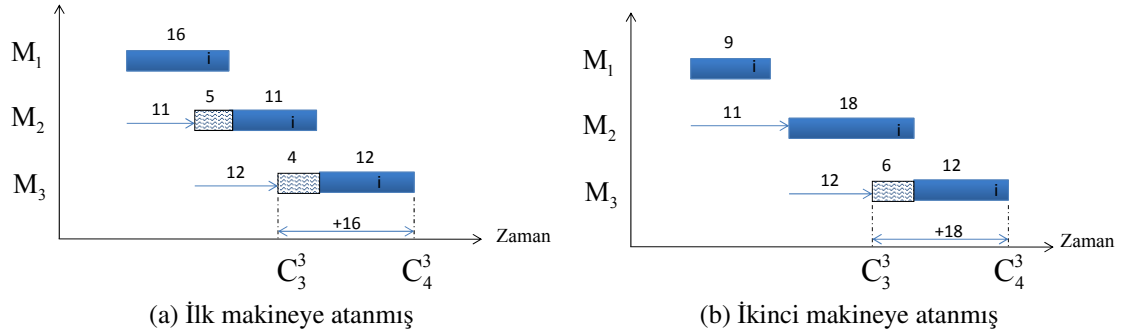


Şekil 5.3: Örnek 4 için beşinci parçanın (11,12) ve (11,13) durumlarındaki Gantt Şemaları

bu şekilde hesaplanabilir.

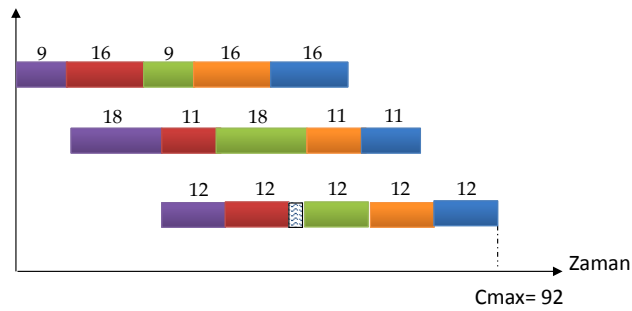
4'üncü aşamaya geçildiğinde, bu parçanın esnek işlemi hem ilk hem de ikinci makineye atanabileceğinden, bu iki durum da değerlendirilmelidir. 4'üncü parçanın esnek işleminin ilk makineye atandığı durumda işlem zamanları 16, 11 ve 12; ikinci makineye atandığı durumda ise 9, 18 ve 12 olacaktır. 4'üncü parçanın esnek işleminin ilk makineye atanması halinde (11,12) durumunda bu parçanın tamamlanma zamanı $r((11, 12), 0) = \max\{0, \max\{0, f^1 + s - a\} + f^2 - b\} + f^3$ formülü kullanılarak 16 (Şekil 5.4a); ikinci makineye atanması halinde ise $r((11, 12), 1) = \max\{0, \max\{0, f^1 - a\} + f^2 + s - b\} + f^3$ formülü yardımı ile 18 (Şekil 5.4b) olarak hesaplanır. Her bir atama kararı verildiğinde hangi aşamaya ulaşılacağını bulmak için $q((a, b)x_i)$ formülleri kullanılmalıdır. 4'üncü parçanın esnek işleminin ilk makineye atandığı durumda ulaşılacak durum, $q((a, b), x_i) = (\max\{f^2, a - (f^1 + s) + f^2\}, \max\{f^3, b - \max\{f^2, a - (f^1 + s) + f^2\} + f^3\})$ formülü yardımı ile (11,12) olarak bulunur. Aynı parçanın esnek işleminin ikinci makineye atandığı durumda ise ulaşılacak durum, $q((a, b), x_i) = (\max\{f^2 + s, a - f^1 + f^2 + s\}, \max\{f^3, b - \max\{f^2 + s, a - f^1 + f^2 + s\} + f^3\})$ formülü kullanılarak, (20,12) olarak bulunur. Çizelgenin bulunması için, her bir parçanın ilk ve ikinci makineye atanması durumlarından en küçük tamamlanma zamanını veren atama durumu seçilmeli ve atama o makineye yapılmalıdır. Bu durum $f_4(11, 12) = \min\{r((11, 12), 0) + f_5(q((11, 12), 0)), r((11, 12), 1) + f_5(q((20, 12), 1))\}$ formülünde yerine konulursa $f_4(11, 12) = \min\{16 + 16, 12 + 18\} = 30$ olarak bulunur ve değere (11,12) durumunda atama ikinci makineye yapılırsa ulaşılır.

Çizelgenin diğer sütunları, bu formüller yardımı ile doldurulur, ilk parçaya ulaşıldığında



Şekil 5.4: Örnek 4 için dördüncü parçanın (11,12) durumundaki Gantt Şemaları

sonraki parçalar için elde edilen toplam tamamlanma zamanı ile ilk parçanın ilk makinedeki tamamlanma zamanı ($C_1^3 = f^1 + f^2 + f^3 + s$) toplandığında optimal yayılma zamanı bulunmuş olur. Bu şekilde hesaplanan optimal çözüm, 1'inci ve 3'üncü parçaların esnek işlemlerinin ikinci; 2'nci, 4'üncü ve 5'inci parçaların esnek işlemlerinin birinci makineye atanması ile 92 olarak bulunmuştur. Optimal çözüme ait Gantt şeması Şekil 5.5'de ve optimal çözüme ulaşmada kullanılan durumlar Çizelge 5.2'de görülmektedir. Bu çizelgede görülen koyu renkle yazılmış değerler optimale ulaşılan değerleri, bu tamamlanma zamanı farkı değerlerinin yanında bulunan parantez içindeki 0-1 değerleri esnek işlemin hangi makineye atanmış olduğunu belirtmektedir. Burada 0 ilk makineye; 1 ise ikinci makineye atandığını göstermektedir.



Şekil 5.5: Örnek 4 için optimal Gantt Şeması

	1	2	3	4	5
(11,12)	x	-	-	32(0)	16(0)
(11,13)	x	-	-	-	15(0)
...	...	-	-	-	-
(12,12)	x	-	-	-	-
(12,13)	x	-	-	-	12(0)
...	...	-	-	-	-
(13,13)	x	53(0)	-	-	-
...	...	-	-	-	-
(17,13)	x	-	-	24(0)	-
...	...	-	-	-	-
(18,12)	92	-	-	-	-
...	...	-	-	-	-
(22,12)	x	-	41(1)	-	-

Çizelge 5.2: Dinamik programlama Örnek 4-optimal çizelge

5.2.2 3 Makineli Sistem, Esnek Makinelerin 2 ve 3 Olduğu Durum ($m = 3, k = 2$)

a ve b 'nin elde edildiği aralıkları bulmak için kullanılacak denklemlerden a 'nın aralığı olan $[f^2, n(f^2 + s - f^1) + f^1]$ değişmezken; b 'nin aralığı $[f^3, n(f^3 + s - f^2) + f^2]$ şeklinde güncellenmiştir. Bu aralıklar da a ve b 'nin alabileceği olası tüm tamsayı değerleri içermektedir. a değerlerinin hesabı için aynı yöntem uygulanmış olup, b için üst sınır değerinden farklı olarak tüm esnek işlemlerin 3'üncü makineye atanmış olma durumunu ele alır. a ve b 'nin alabileceği en büyük değerler $m = 3, k = 1$ problemi için anlatılan yaklaşımla hesaplanmaktadır.

i 'inci aşama $S_i = (a, b)$ durumunda elde edilecek tamamlanma zamanları farkı aşağıdaki formüllerle hesaplanabilir:

- Esnek işlemin ikinci makineye atanmış ise ($x_i = 0$)

$$r((a, b), 0) = \max\{0, \max\{0, f^1 - a\} + f^2 + s - b\} + f^3$$

- Esnek işlem üçüncü makineye atanmış ise ($x_i = 1$),

$$r((a, b), 1) = \max\{0, \max\{0, f^1 - a\} + f^2 - b\} + f^3 + s$$

i aşamasında $S_i = (a, b)$ durumunda verilen x_i kararına göre ulaşılabilecek (\hat{a}, \hat{b})

değerlerini hesaplamak için aşağıdaki formüller kullanılabilir:

- Esnek işlem ilk makineye atandığında ($x_i = 0$) yeni durum hesabı aşağıdaki gibi yapılabilir:

$$q((a, b), 0) = (\hat{a}, \hat{b}) = (\max\{f^2, a - f^1 + f^2 + s\}, \max\{f^3, b - \max\{f^2, a - f^1 + f^2 + s\} + f^3\})$$

- Esnek işlemin ikinci makineye atandığı durumda ($x_i = 1$) ise yeni durumun hesaplanması için aşağıdaki formüller kullanılmalıdır:

$$q((a, b), 1) = (\hat{a}, \hat{b}) = (\max\{f^2, a - f^1 + f^2\}, \max\{f^3, b - \max\{f^2 + s, a - f^1 + f^2\} + f^3 + s\})$$

(\hat{a}, \hat{b}) için kullanılacak denklemler ise x_i 'nin alacağı değerler göz önüne alınarak şu şekilde özetlenebilir $\hat{a} = \max\{f^2, a - f^1 + f^2 + s.(1 - x_i)\}$ ve $\hat{b} = \max\{f^3, b - \max\{f^2, a - f^1 + f^2 + s.(1 - x_i)\} + f^3 + s.x_i\}$

$m = 3, k = 2$ problemi için yenileme denklemi aşağıdaki gibidir:

$$f_i(a, b) = \min_{x_i \in \{0,1\}} \{(r(a, b), x_i) + f_{i+1}(q((a, b), x_i))\}$$

Sonuç olarak, $i = n - 1$ 'den başlanarak $i = 1$ 'e kadar geriye dönük hesaplamalar yapılır. $m = 3, k = 2$ problemi için geliştirilmiş dinamik programlama yönteminin çözüm süresi $O((n^3(f^2 - f^1)(f^3 - f^2) + n^2(f^2 - f^1)(ns + 1) + n^2(f^3 - f^2)(ns + 1) + n(ns + 1)^2)$ 'dir. Bu çözüm süresi de a ve b değerlerinin içinde olduğu aralıkların boyutu ile işlem yapılacak toplam parça sayısı $n - 1$ 'in çarpılması sonucu elde edilmiştir.

5.2.3 4 Makineli Sistem, Esnek Makinelerin 2 ve 3 Olduğu Durum ($m = 4, k = 2$)

Önceki bölümde anlatılan 3 makineli sistemde, esnek makineler 1-2 veya 2-3 olabilir. Ancak bir sistemde esnek makineler sadece ilk veya son makineler olmak zorunda değildir; esnek makinelerin önünde ve arkasında bazı esnek olmayan ve önteorom 5'teki gibi yoksayılmayacak makineler olabilir. Bu nedenle esnek olmayan makine sürelerinin ve konumlarının sisteme etkisinin anlaşılması için, m makineli sisteme geçişte incelenmesi gereken basamaklardan biri de 4 makineli sistemdir.

4 makineli bir esnek akış tipi sistem ele alalım. Esnek makinelerin öncesinde ve sonrasında olabilecek esnek olmayan makineleri de temsil edebilmesi açısından, esnek makineler 2'nci ve 3'üncü makineler olsun. Bu şartlar altında oluşabilecek durumlar, tamamlanma zamanları fark formülleri ve yenileme denklemi aşağıdaki gibi olacaktır:

4 makineli sistemde karşılaşılabilecek durumlar da her bir parçanın ardışık makineler arasındaki tamamlanma zamanları farklarını içerecektir. Aynı parçanın (herhangi bir i parçası) makineler arasındaki tamamlanma zamanı farklarının alabileceği değerler a , b ve c ile gösterilsin ($a = C_i^2 - C_i^1$, $b = C_i^3 - C_i^2$ ve $c = C_i^4 - C_i^3$).

Buna göre a , $[f^2, \max\{f^2, n(f^2 - f^1) + f^1\}]$ aralığında herhangi bir tamsayı değerini alabilir. Bu aralıktaki en küçük değer esnek işlemin üçüncü makineye atanması ile ikinci makinedeki işlem süresinin f^2 değerini alırken, en büyük değer tüm parçaların esnek işlemlerinin üçüncü makineye atanması durumunda oluşabilecek en büyük farkı göstermektedir. b , $[f^3, n(f^3 + s - f^2) + f^2]$ aralığındaki ve c , $[f^4, \max\{f^4, n(f^4 - f^3) + f^3 + s\}]$ aralığındaki tüm tamsayı değerlerini alabilir. Sonuç olarak, ortaya çıkabilecek bütün (a, b, c) üçlü durumların sayısı, bu aralıklarda yer alan tamsayı değerlerinin çarpımına eşittir. Buradaki a ve b değerleri önceki bölümlerde bahsedildiği gibi hesaplanırken, c 'nin en küçük değeri 4'üncü makinenin sabit işlem zamanına (f^4) eşittir ve en büyük değeri ise ilk parça hariç tüm esnek işlemlerin 2'nci makineye atanması ile bulunur. Burada a ve b değerlerinin hesaplanması $m = 3, k = 1$ durumu ile aynı yaklaşımı kullanmaktadır.

i 'inci aşama $S_i = (a, b)$ durumunda tamamlanma zamanları farkı formülleri aşağıdaki gibi hesaplanabilir:

- Esnek işlem ikinci makineye atanmışsa ($x_i = 0$),

$$r((a, b), 0) = \max\{0, \max\{0, \max\{0, f^1 - a\} + f^2 + s - b\} + f^3 - c\} + f^4$$

- Esnek işlem üçüncü makineye atandığında ise ($x_i = 1$),

$$r((a, b), 1) = \max\{0, \max\{0, \max\{0, f^1 - a\} + f^2 - b\} + f^3 + s - c\} + f^4$$

şeklinde hesaplanabilir.

$S_i = (a, b, c)$ durumunda verilen karara göre $S_{i+1} = (\hat{a}, \hat{b}, \hat{c})$ değerleri için aşağıdaki $q((a, b, c), x_i)$ formülleri kullanılabilir:

- Esnek işlemin ikinci makineye atandığı durumda ($x_i = 0$),

$$q((a, b, c), 0) = (\hat{a}, \hat{b}, \hat{c}) = (\max\{f^2, a - f^1 + f^2 + s\}, \max\{f^3, b - \max\{f^2, a - f^1 + f^2 + s\} + f^3\}, \max\{f^4, c - \max\{f^3, b - \max\{f^2, a - f^1 + f^2 + s\} + f^3\} + f^4\})$$

- Esnek işlem üçüncü makineye atandığı durumda ($x_i = 1$),

$$q((a, b, c), 1) = (\hat{a}, \hat{b}, \hat{c}) = (\max\{f^2, a - f^1 + f^2\}, \max\{f^3, b - \max\{f^2, a - f^1 + f^2\} + f^3 + s\}, \max\{f^4, c - \max\{f^3, b - \max\{f^2, a - f^1 + f^2\} + f^3 + s\} + f^4\})$$

$(\hat{a}, \hat{b}, \hat{c})$ için kullanılacak denklemler yeniden düzenlenerek şu şekilde özetlenebilir:

$$\hat{a} = \max\{f^2, a - f^1 + f^2 + s \cdot (1 - x_i)\}, \hat{b} = \max\{f^3, b - \max\{f^2, a - f^1 + f^2 + s \cdot (1 - x_i)\} + f^3 + s \cdot x_i\} \text{ ve}$$

$$\hat{c} = \max\{f^4, c - \max\{f^3, b - \max\{f^2, a - f^1 + f^2 + s \cdot (1 - x_i)\} + f^3 + s \cdot x_i + f^4\}\}$$

Sonuç olarak $m = 4, k = 2$ problemi için yenileme denklemi aşağıdaki gibidir:

$$f_i(a, b) = \min_{x_i \in \{0,1\}} \{(r(a, b, c), x_i) + f_{i+1}(q((a, b, c), x_i))\}$$

4 makine için yazılmış tüm bu formüller, bir önceki bölümde anlatılan 3 makine için yazılmış ifadelerle aynı mantığı taşımaktadır. Amaç, her bir parça için her makinedeki tamamlanma zamanlarını tespit ederek son makinedeki tamamlanma zamanı ile farkını belirleyebilmektir. Bunun yanısıra hangi parçanın esnek işleminin hangi makineye atandığını bularak çizelgenin de elde edilmesi istenmektedir. $m = 4, k = 2$ problemi için geliştirilmiş bu dinamik yöntemin çözüm süresi $O(\max\{n, n^4(f^2 - f^1)(f^3 - f^2)(f^4 - f^3) + n^3(f^2 - f^1)(ns + 1)(f^4 - f^3)\})$ 'dir. Bu çözüm süresi de a, b ve c değerlerinin içinde olduğu aralıkların boyutu ile $n - 1$ 'in çarpılması ile elde edilmiştir.

4 makineli bir sistemde esnek makinelerin ilk veya son makineler olduğu durum, yapılan indirgeme işlemleri ile birlikte 3 makineli sistem haline dönüşebilmektedir. Bu dönüşüm bir sonraki bölümde detaylı bir şekilde anlatılacaktır.

5.2.4 m Makineli Sistem

m makineli sistem çözümü için, 3 ve 4 makine başlıklarında incelenen, optimallik koşulları belirlenmiş sistemlerden yararlanılmaktadır. m makineli sistem önteoremler

yardımı ile çeşitli adımlar izlenerek 3 veya 4 makineli sisteme indirgenmekte, önceki bölümlerde anlatılan çözüm yaklaşımları kullanılarak optimal sonuç bulunmaktadır.

m makineli sisteme geçişte incelenmesi gereken ilk durum problemin Önteorem 3'e ve Önteorem 4'e uygun olup olmadığıdır. Bu iki önteoremden herhangi bir tanesine uygunsa, atama kolaylıkla bulunur; uygun değilse m makineli dinamik programlama adımlarına geçilir.

Teorem 1 *Sınırsız ara stok kapasiteli sistemlerde m makineli herhangi bir problem 3 makine $k = 1$, 3 makine $k = 2$ veya 4 makine $k = 2$ sistemlerinden birine indirgenebilir.*

İspat: Problem verileri incelenir. Esnek makineler haricindeki makinelerin sabit işlem zamanları gözönüne alınarak aşağıdaki indirgeme adımları uygulanır:

1. Önteorem 5'ten yararlanılarak, esnek olmayan makinelerin sabit işlem zamanları, esnek makinelerin sabit işlem zamanlarından küçük olan makineler yoksayılır. Sonraki adımlarda yapılacak işlemlerde bu makineler ilk parçanın ataması haricinde hesaba katılmaz, makine sayısı için ilk indirgeme yapılmış olur.

2. *Esnek makineler 1 ve 2 numaralı makineler ise,*

İlk indirgeme sonucu kalan makinelerden esnek makineler hariç en büyük sabit işlem zamanlı olan makine seçilir. (Bu makine l makinesi ve sabit işlem zamanı da f^l olarak adlandırılacaktır.)

2.1. *En büyük işlem zamanlı esnek olmayan l makinesi esnek makinelerin arkasındaki makine (3.makine) ise;*

Önteorem 1'e göre problemin tersi alınır. Yeni durumda $(m - 2)$ 'nci makine olan l makinesinden önceki makineler $(1, \dots, m - 1)$ yoksayılır. Çünkü, daha küçük işlem zamanlı makinelerin l makinesinde parçaların işlenmesine bir etkisi yoktur. Önceki makinelerin işlem zamanları f^l 'den küçük olduğu için l makinesinde herhangi bir boş zaman oluşmayacaktır. Bu şekilde indirgenmiş sistem için bulunan çözüm orijinal sisteme kolayca adapte edilebilir. Aşağıdaki örnek, yöntemin anlaşılmasını kolaylaştıracaktır.

2.2. *En büyük sabit işlem zamanlı esnek olmayan l makinesi ile esnek makineler arasında bir veya daha fazla esnek olmayan makine varsa;*

Önteorem 1'e göre problemin tersi alınır. l makinesinden önceki daha küçük sabit işlem zamanına sahip makineler kaldırılır. l makinesi, esnek makineler ile arasında bulunan makine ya da makineleri onlardan daha büyük işlem zamanına sahip olduğu için domine eder. Aradaki makinelerin işlem zamanları $f^j + (f^l - f^j)$ şeklinde, yani f^l cinsinden yazılabilir. Bu yüzden, ilk parça yerleştirildikten sonra $(k-1)$ 'inci makineye kadar olan makineler ihmal edilir, $(k-1)$ 'inci makinenin işlem zamanı f^l cinsinden yazılır. Böylece indirgenmiş problemde, işlem zamanları f^j ile boş zamanlar sabit bir şekilde $(f^l - f^j)$ olur. İlk makinedeki ilk parçanın başlangıç zamanı $f^1 + \dots + f^{k-2}$ 'dir. Makine azaltma ile 3 makineli sistem haline dönüşen bu problemi, $m = 3, k = 2$ sistemi için yapılmış denklemler ile çözmek mümkün olur.

3. *Esnek makineler $m - 1$ ve m numaralı makineler ise,*

Önteorem 1'e göre problemin tersi alınarak ilk iki makine esnek makine haline getirilir. Önceki adımda belirtilen yöntemlerden, problem parametrelerine uygun olan yöntem seçilerek çözüm bulunur.

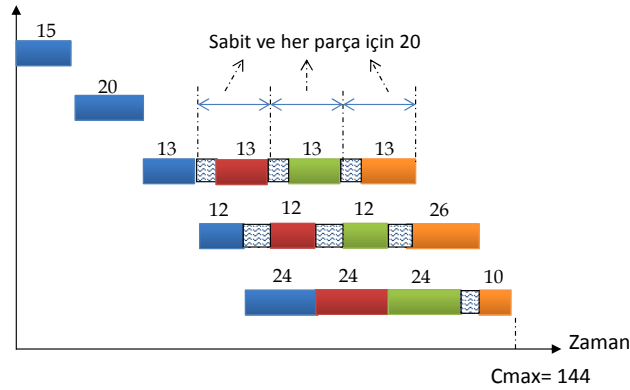
4. *Esnek makineler ilk ya da son makineler değilse,*

Esnek makinelerin önünde veya arkasında makineler olduğu durumdur. Bu durumda hem önündeki hem de arkasındaki makineler için en büyük işlem zamanlı makineler seçilir. Bu makineler ile esnek makineler arasında makine yoksa sistem doğrudan 4 makineli sisteme indirgenmiş olur. Eğer arada başka makineler varsa, 2.2'nci maddede belirtildiği gibi ikinci bir makine indirgenmesi yapılır. 4 makineli sisteme dönüştürülen problem, uygun yöntem seçilerek çözülür. \square

Örnek 5 Adım 2.1 yardımı ile çözülebilecek bir problemin parametreleri $n = 4, m = 5, k = 1, f^1 = 10, f^2 = 12, f^3 = 20, f^4 = 13, f^5 = 15, s = 14$ şeklinde olabilir. Bu durumda problemin tersini düşündüğümüzde 13 ve 15 işlem sürelerine sahip makineler yoksayılar. Problem sabit işlem süreleri sırayla 20, 12 ve 10, $n = 10, s = 14, m = 3, k = 2$ olan bir sisteme dönüşür. Bu problem $m = 3, k = 2$ için geliştirilmiş yöntemle çözülebilir.

Örnek 6 Adım 2.2 yardımı ile çözülebilecek bir örnekte problem verileri $n = 4, m =$

5, $k = 1$, $f^1 = 10$, $f^2 = 12$, $f^3 = 13$, $f^4 = 20$, $f^5 = 15$, $s = 14$ şeklinde verilmiş olsun. Bir önceki adımda bahsedilen özellik kullanılarak tersi alınır ve 15 işlem süresine sahip makine yoksayılır, böylece problem 4 makineye indirilmiş olur. 20, 13, 12 ve 10 işlem zamanları ile sıralanan makinelerde 2'nci makinenin 1'inci makine tarafından domine edildiği görülür. 2'nci makinenin işlem süresi ve boş zaman durumu öncesinde makine varmış gibi yapılır. Parçalar sonrasındaki makinelerde bu süreler göre çizelgelenir. Bu şekle dönüşen probleme $m = 3$, $k = 2$ yöntemleri ile çözüm bulunabilir. Şekil 5.6'te de görüldüğü gibi, en uzun işlem süresine sahip olan 1'inci makinede ve işlem süresi küçük olduğu için yoksayılan 15 sabit işlem süreli makinede sadece ilk parçanın ataması yapılmıştır. En büyük sabit işlem süreli makineye sadece ilk parçanın ataması yapılmış olmasına rağmen, bu makinenin daha küçük işlem süreli makinelere etkisi devam etmektedir. Örneğin, 3'üncü makinedeki boş zaman miktarları esnek makinelerden önceki en uzun işlem süresi olan 20'ye göre belirlenmiştir.



Şekil 5.6: Örnek 6 için optimal Gantt Şeması

Örnek 7 Adım 4 yardımı ile çözülebilecek bir problem verileri $n = 4$, $m = 7$, $k = 4$, $f^1 = 9$, $f^2 = 10$, $f^3 = 20$, $f^4 = 10$, $f^5 = 12$, $f^6 = 15$, $f^7 = 8$, $s = 14$ şeklinde verilmiş olsun. f^4 ve f^5 'ten küçük işlem zamanlı olan birinci ve ikinci makine 2.1'inci maddeye göre yoksayılır. Problem ters çevrilerek f^4 ve f^5 'ten küçük işlem süresine sahip olan ilk makine de 2.1'inci maddeye göre yoksayılır. Problem 4 makineye indirgenmiş olur. 4 makineli bu sistemde, ilk ve son makinelerde bir indirgenme işlemi yapılamadığı için, $f^1 = 15$, $f^2 = 12$, $f^3 = 10$, $f^4 = 20$ verilerine sahip bu problem $m = 4$, $k = 2$ için geliştirilmiş çözüm yöntemi kullanılarak çözümler.

m makineli sistem, çoğu zaman 3 makineli sisteme dönüştürülüp çözülebilir. Bu

durumda çözüm süresi çok daha kısa olur. Ancak m makineden 4 makineye kadar indirilmiş esnek işlemlerin 2'nci ve 3'üncü makineler olduğu bir problemin sabit ve esnek işlem süreleri, $f^1 > f^2$, $f^4 > f^3$, $f^1 < f^2 + s$, $f^4 < f^3 + s$ koşullarına aynı anda sahip olursa, bu sistemi 3 makineli sisteme indirgemek mümkün değildir, 4 makineli olarak çözmek gerekir. Bu problem türü için optimal sonuca ulaşılmak istendiğinde, incelediği durum sayısının fazla olması nedeni ile çok daha uzun sürede çözüm verecek olan bu yöntem seçilmelidir.

5.3 Sıfır Ara Stok Problemi

Sıfır ara stok probleminin çözümünde, sınırsız ara stok problemine benzer yaklaşım uygulanmıştır. Ancak, her bir parçanın ardışık makinelerdeki tamamlanma zamanları farkları olan ve durumları belirleyen a, b, c değerleri ile minimum tamamlanma zamanı farklarının hesaplanmasında çeşitli farklılıklar bulunmaktadır. Ayrıca sınırsız ara stok kapasiteli problemlerden farklı olarak i parçasının k ve $k + 1$ 'inci makinelere atanma durumunun yanı sıra $i - 1$ 'inci parçanın da k ve $k + 1$ 'inci makinelere atanma durumu göz önüne alınarak formüller geliştirilmiştir.

5.3.1 3 Makineli Sistem, Esnek Makinelerin 1 ve 2 Olduğu Durum

$$(m = 3, k = 1)$$

Ara stok kapasitesinin sıfır olduğu sistemlerde makineler işlemini bitirdikleri parçayı sonraki makineye vermek için o makinedeki işlemin bitmesini beklemek zorundadırlar. Bu yüzden ilk makinede i parçasının başlangıç zamanı, ikinci makinedeki $i - 1$ parçasının başlangıç zamanına eşit olmalıdır ($T_i^1 = T_{i-1}^2$). Bu özellik, esnek işlemlerin her bir makineye atanması göz önüne alındığında 4 farklı durum oluşturur ($(x_{i-1}^1, x_i^1) = (0, 0), (0, 1), (1, 0), (1, 1)$). Bu durumlar sonucu oluşabilecek a değerleri aşağıdaki şekildedir:

- $2f^2 + s - f^1$
- $\max\{f^2 - f^1, 0\} + b - f^3 + f^2$
- $2f^2 - f^1 - s$

- f^2
- $f^2 + s$
- $2f^2 - f^1 + 2s$
- $\max\{f^2 - f^1, 0\} + b - f^3 + f^2 + s$
- $2f^2 - f^1$

Problem verilerine bağlı olarak bu değerlerden bazıları, negatif ya da f^2 'den küçük olabilmektedir. a değerinin en az f^2 'ye eşit olduğu için negatif ya da f^2 'den küçük değerler hesaplamaya dahil edilmeyecektir. b 'nin alabileceği en düşük değer üçüncü makinedeki sabit işlem zamanına eşitken; en büyük değer bütün esnek işlemlerin ilk makineye atandığı durumda $n(f^3 - f^2) + f^2 + s$ olarak hesaplanır. Dolayısıyla b , $[f^3, \max\{f^3, n f^3 - n f^2 + f^2 + s\}]$ aralığındaki bütün tamsayı değerlerini alabilir.

i 'inci aşama $S_i = (a, b)$ durumunda, esnek işlemin ilk ve ikinci makineye atanması durumlarında ($x_i = 0$ ve $x_i = 1$) bu işin tamamlanma zamanı formülünü elde edebilmek için aşağıdaki tamamlanma zamanları hesaplamaların yapılmış olması gerekir. Bu hesaplar i parçasının atanmasının yanında sınırsız ara stok probleminden farklı olarak $i - 1$ parçasının atanmasını da değerlendirerek tamamlanma zamanı hakkında bilgiye ulaşılmasını sağlar.

i 'inci aşama $S_i = (a, b)$ durumu için esnek işlemin ilk makineye atandığı durumda ($x_i = 0$);

- $i - 1$ parçasının esnek işlemi ilk makineye atanmışsa ($x_{i-1} = 0$) tamamlanma zamanları aşağıdaki gibi hesaplanır:

$$C_i^1 = a - f^2 + f^1 + s, C_i^2 = \max\{C_i^1, a + b - f^3\} + f^2, C_i^3 = \max\{C_i^2, a + b\} + f^3$$

- $i - 1$ parçasının esnek işlemi ikinci makineye atandığı durumda ($x_{i-1} = 1$) ise tamamlanma zamanları aşağıdaki gibi olmaktadır:

$$C_i^1 = a - f^2 + f^1, C_i^2 = \max\{C_i^1, a + b - f^3\} + f^2, C_i^3 = \max\{C_i^2, a + b\} + f^3$$

i aşaması $S_i = (a, b)$ durumu için esnek işlemin ikinci makineye atanması durumunda ise ($x_i = 1$);

- $i - 1$ parçasının esnek işleminin ilk makineye atanırsa ($x_{i-1} = 0$) tamamlanma zamanları aşağıdaki gibi hesaplanır:

$$C_i^1 = a - f^2 + f^1, C_i^2 = \max\{C_i^1, a + b - f^3\} + f^2 + s, C_i^3 = \max\{C_i^2, a + b\} + f^3$$

- $i - 1$ parçasının esnek işleminin ikinci makineye atandığı durumda ($x_{i-1} = 1$) ise tamamlanma zamanları aşağıdaki gibi hesaplanmaktadır:

$$C_i^1 = a - f^2 - s + f^1, C_i^2 = \max\{C_i^1, a + b - f^3\} + f^2 + s, C_i^3 = \max\{C_i^2, a + b\} + f^3$$

Tüm bu hesaplamalar i 'inci aşamadaki tamamlanma zamanları farklarını hesaplamak için birer basamaktır. Her bir parçanın $x_i = 0$ ve $x_i = 1$ durumları için tamamlanma zamanı farkları aşağıdaki formülle hesaplanabilir:

$$r((a, b), x_i) = C_i^3 - (a + b)$$

Burada i ve $i - 1$ 'inci parçaların ayrı ayrı k ve $k + 1$ 'inci makinelere atanma durumları incelendiği için k ve $k + 1$ 'e atanma durumlarının ayrı ayrı formüller halinde tekrar ifade edilmesine gerek yoktur, bu bilgi formüllerle hesaplanmış C_i^3 'ün içinde yer almaktadır.

$m = 3, k = 1$ sıfır ara stok problemi için yenileme denklemi aşağıdaki şekilde ifade edilebilir:

$$f_i(a, b) = \min_{x_i \in \{0,1\}} \{r((a, b), x_i) + f_{i+1}(q((a, b), x_i))\}$$

$S_i = (a, b)$ durumunda oluşacak $S_{i+1} = (\hat{a}, \hat{b})$ durumunu hesaplamak için i parçasının hangi makineye atanmış olduğunun bilinmesi gerekmektedir. Bu durum göz önüne alındığında yukarıda yapılmış olan tamamlanma zamanı hesapları dikkate alınarak şu hesaplamalar yapılabilir:

$$q((a, b), x_i) = (C_i^2 - C_i^1, C_i^3 - C_i^2)$$

$m = 3, k = 1$ sistemi sıfır ara stok problemi için geliştirilmiş bu yöntemin çözüm süresi $O(\max\{n, n^2(f^3 - f^2)\})$ 'dir.

5.3.2 3 Makineli Sistem, Esnek Makinelerin 2 ve 3 Olduğu Durum ($m = 3, k = 2$)

Esnek makinelerin 2 ve 3 olduğu durumda başlangıç zamanları arasında işlem kolaylığı sağlayabilecek herhangi bir bağıntı bulunmamaktadır. Bu yüzden $a, [f^2, \max\{f^2 + f^3 + 2s - f^1, 2f^2 + 2s - f^1\}]$ aralığındaki tüm tamsayı değerlerini alabilirken, b 'nin alabileceği değerler $[f^3, 2f^3 + 2s - f^2]$ aralığındaki tamsayılardır.

$S_i = (a, b)$ durumu i aşaması için esnek işlemin ikinci makineye atandığı durumda ($x_i = 0$);

- $i-1$ parçasının esnek işlemi de ikinci makineye atanırsa ($x_{i-1} = 0$) tamamlanma zamanları aşağıdaki gibi hesaplanır:

$$C_i^1 = a - f^2 - s + f^1, C_i^2 = \max\{C_i^1, a + b - f^3\} + f^2 + s, C_i^3 = \max\{C_i^2, a + b\} + f^3$$

- $i-1$ parçasının esnek işleminin üçüncü makineye atandığı durumda ($x_{i-1} = 1$) ise tamamlanma zamanları aşağıdaki gibi olmaktadır:

$$C_i^1 = a - f^2 + f^1, C_i^2 = \max\{C_i^1, a + b - f^3 - s\} + f^2 + s, C_i^3 = \max\{C_i^2, a + b\} + f^3$$

$S_i = (a, b)$ durumu i aşaması için esnek işlemin üçüncü makineye atanması durumunda ise ($x_i = 1$);

- $i-1$ parçasının esnek işlemi ikinci makineye atanırsa ($x_{i-1} = 0$) tamamlanma zamanları aşağıdaki gibi hesaplanır:

$$C_i^1 = a - f^2 - s + f^1, C_i^2 = \max\{C_i^1, a + b - f^3 - s\} + f^2 + s, C_i^3 = \max\{C_i^2, a + b\} + f^3$$

- $i-1$ parçasının esnek işleminin üçüncü makineye atandığı durumda ($x_{i-1} = 1$) ise tamamlanma zamanları aşağıdaki gibi hesaplanmaktadır:

$$C_i^1 = a - f^2 + f^1, C_i^2 = \max\{C_i^1, a + b - f^3 - s\} + f^2, C_i^3 = \max\{C_i^2, a + b\} + f^3 + s$$

$i-1$ parçası ile i parçasının tamamlanma zamanları arasındaki fark, $x_i = 0$ ve $x_i = 1$ durumları için de aşağıdaki gibi hesaplanmaktadır:

$$r((a, b), x_i) = C_i^3 - (a + b)$$

$S_i = (a, b)$ durumunda i parçasının atmasına göre oluşacak $S_{i+1} = (\hat{a}, \hat{b})$ durumu hesabı aşağıdaki gibi yapılabilir:

$$q((a, b), x_i) = (\hat{a}, \hat{b}) = (C_i^2 - C_i^1, C_i^3 - C_i^2)$$

$m = 3, k = 2$ sıfır ara stok problemi için geliştirilen yöntemin çözüm süresi, a 'nın alabileceği en büyük değere göre değişmekle birlikte, $O(\max\{n(f^2)^2 + nf^1f^2 + 4ns^2, nf^2f^3 + 4ns^2 + nf^1f^2\})$ olmaktadır.

5.3.3 4 Makineli Sistem, Esnek Makinelerin 2 ve 3 Olduğu Durum ($m = 4, k = 2$)

4 makineli sistem için de sınırsız ara stok kapasiteli problemde olduğu gibi esnek makinelerin 2 ve 3'üncü makineler olduğu durum ele alınmıştır. Burada da S_i durumu a, b, c şeklinde üçlü olarak ifade edilmektedir. Ancak, bu değerlerin hesaplanmasında çeşitli farklılıklar bulunmaktadır. a değeri $[f^2, \max\{f^2 + f^3 + 2s - f^1, 2f^2 + 2s - f^1\}]$ aralığındaki; b değeri $[f^3, 2f^3 + 2s - f^2]$ aralığındaki ve c değeri $[f^4, nf^4 - nf^3 + f^3]$ aralığındaki tamsayıları alabilmektedir.

$S_i = (a, b, c)$ durumu i aşaması için esnek işlemin ikinci makineye atandığı durumda ($x_i = 0$);

- $i - 1$ parçası ikinci makineye atanırsa ($x_{i-1} = 0$) tamamlanma zamanları

$$C_i^1 = a - f^2 - s + f^1, C_i^2 = \max\{C_i^1, a + b - f^3\} + f^2 + s, C_i^3 = \max\{C_i^2, a + b\} + f^3, C_i^4 = \max\{C_i^3, a + b + c\} + f^4$$

şeklinde hesaplanır.

- $i - 1$ parçası üçüncü makineye atanırsa ($x_{i-1} = 1$) tamamlanma zamanları

$$C_i^1 = a - f^2 + f^1, C_i^2 = \max\{C_i^1, a + b - f^3 - s\} + f^2 + s, C_i^3 = \max\{C_i^2, a + b\} + f^3, C_i^4 = \max\{C_i^3, a + b + c\} + f^4$$

formülleri ile bulunabilir.

$S_i = (a, b, c)$ durumu i aşaması için esnek işlemin üçüncü makineye atandığı durumda ($x_i = 1$);

- $i - 1$ parçası ikinci makineye atanırsa ($x_{i-1} = 0$) tamamlanma zamanları

$$C_i^1 = a - f^2 - s + f^1, C_i^2 = \max\{C_i^1, a + b - f^3\} + f^2, C_i^3 = \max\{C_i^2, a + b\} + f^3 + s, C_i^4 = \max\{C_i^3, a + b + c\} + f^4$$

formülleri ile hesaplanır.

- $i - 1$ parçası üçüncü makineye atanırsa ($x_{i-1} = 1$) tamamlanma zamanları

$$C_i^1 = a - f^2 + f^1, C_i^2 = \max\{C_i^1, a + b - f^3\} + f^2, C_i^3 = \max\{C_i^2, a + b\} + f^3 + s, C_i^4 = \max\{C_i^3, a + b + c\} + f^4$$

şeklinde hesaplanabilir.

i 'inci aşamada atamaya bağlı olarak elde edilecek $i - 1$ ve i parçalarının tamamlanma zamanları arasındaki fark hesabı aşağıdaki şekilde yapılır:

$$r((a, b, c), x_i) = C_i^4 - (a + b + c)$$

i parçasının atamasına göre $S_i = (a, b)$ durumunda oluşacak $S_{i+1} = (\hat{a}, \hat{b}, \hat{c})$ durumu hesabı aşağıdaki gibi yapılabilir:

$$q((a, b, c), x_i) = (\hat{a}, \hat{b}, \hat{c}) = (C_i^2 - C_i^1, C_i^3 - C_i^2, C_i^4 - C_i^3)$$

$m = 4, k = 2$ problemi sıfır ara stok kapasiteli durum için geliştirilen yöntemin çözüm süresi, a 'nın alabileceği en büyük değer ile değişmekle birlikte, $O((f^3)^2 f^2 n^2 + (f^3)^2 f^4 n^2 + 4s^2 n^2 f^4 + f^1 f^2 n^2 f^4)$ ile $O((f^2)^2 n^2 f^3 + 4s^2 n^2 f^4 + f^1 f^2 n^2 f^4)$ değerlerinden büyük olana eşittir.

5.3.4 m Makineli Sistem

Teorem 2 Sıfır ara stok kapasiteli m makineli sistem, sınırsız ara stok kapasiteli sistem ile benzer şekilde, 3 makineli $k = 1$, 3 makineli $k = 2$ veya 4 makineli $k = 2$ sistemlerine indirgenebilir.

İspat: m makineli sınırsız ara stok problemi için detaylı bir şekilde anlatılmış çözüm yöntemi m makineli sıfır ara stok problemi için de geçerlidir. Önceki bölümlerde

anlatılan 3 ve 4 makineli sistemler için düzenlenmiş a, b, c aralıkları ve tamamlanma zamanı fark formülleri kullanılarak çözüm yöntemi uygulanırsa optimal sonuç elde edilir. \square

Sınırsız veya sıfır ara stok kapasiteli problemler için geliştirilmiş dinamik programlama yöntemlerinin çözüm süreleri yapay polinom zamanlıdır. Problem parametrelerine göre çözüm süreleri değişmektedir. Ancak hem sınırsız hem de sıfır ara stok kapasiteli problemler için 4 makineli sistemlerin 3 makineli sistemlere göre daha uzun sürede çözüm verdiği görülmektedir. Ancak 3 makineli sistemler arasında bu şekilde bir karşılaştırma yapılamamaktadır.

Geliştirilmiş dinamik programlama yöntemlerinin optimal sonuç verdiği bilinmekle birlikte, problem parametrelerinin bazı değerleri için çözüm süresi oldukça uzun olmaktadır. 3 ve 4 makineli sistemlerin çözüm zamanları işlem sürelerine bağlı yapay polinom zamanlıdır. Problemin çözüm süresi parça sayısının yanı sıra sabit ve esnek işlem sürelerine (f^j, s) de bağlıdır. Bu parametreler 10, 20 gibi değerler aldığı anda çözüm süresi çok kısa olurken $10^6, 10^9$ gibi değerler olması durumunda çözüm süreleri oldukça uzayacaktır. Gerçek hayatta bu kadar uzun işlem sürelerine rastlanmayacağı düşünülse de f^j ve s 'in tamsayı oldukları varsayımı sonucu, tamsayı olmayan değerleri tamsayıya çevirmek için yapılacak bir katsayı ile çarpma işlemi bu büyüklükte bir değer çıkmasına neden olabilecektir. Problemi optimal çözmek önemlidir. Ancak rekabetçi çevrede zaman kısıtını da dikkate almak gerekir. Bu yüzden, dinamik çözüm yönteminden daha kısa sürede iyi sonuçlar verecek bir sezgisel çözüm yöntemi geliştirilmiştir. Sonraki bölümde anlatılacak bu sezgisel yöntem, oldukça kısa sürelerde ve optimalden binde seviyelerinde farklı sonuçlar vermektedir. Problem parametrelerinin uygun olduğu ve kesin sonuç elde edilmesi kaçınılmaz olan durumlarda dinamik çözüm yönteminin, diğer durumlarda ise sezgisel çözüm yönteminin kullanılması önerilmektedir.

6. Sezgisel Çözüm Yöntemi

Bu bölümde, ele alınan sınırsız veya sıfır ara stok kapasiteli problemler için geliştirilmiş polinom zamanlı sezgisel çözüm yönteminin uygulama basamakları, çeşitli örnekler, sezgisel yöntemin çözüm kalitesinin testi için yapılan çalışmalar, sonuçları ve değerlendirmeler sunulmuştur.

6.1 Algoritma Adımları

Yardımcı teoremlerden yararlanarak problemin çözümü için sezgisel bir algoritma geliştirilmiştir. Algoritma adımları aşağıdaki şekilde ifade edilebilir.

Adım 1: 2 parça ile algoritmaya başlanır. Önteorem 3'te gösterildiği üzere ilk parçanın esnek işlemi ikinci esnek makineye; ikinci parçanın esnek işlemi ise ilk esnek makineye atanır.

Adım 2: Mevcut çizelgeye yeni bir parça eklenir. Bu parça önceden çizelgelenmiş parçaların aralarına yerleştirilmeye çalışılır. Her alternatif yerleştirme için esnek işlemin ilk ve ikinci esnek makinelere atanma alternatifleri ayrı ayrı değerlendirilir. Bu alternatiflerden tamamlanma zamanı küçük olan seçilir.

Adım 3: Bütün parçalar yerleştirilene kadar bir önceki adımda yapılanlar tekrarlanır.

Bu algoritma, her parçayı yerleştirirken, mevcut çizelgedeki bütün parçaların aralarına yerleştirme alternatifini dener. Bu işlemi n parçanın hepsi için yapacağından en kötü durumda $n - 1$ tane alternatif dener. Bu yüzden geliştirilen algoritmanın çözüm süresi $O(n^2)$ 'dir.

Sınırsız ve sıfır ara stok problemi için geliştirilen algoritmanın adımları aşağıdaki şekildedir:

Algoritma 1

Girdi: $n, m, k, s, f^j, \forall j$

Çıktı: $T_i^j, C_i^j, x_i^k, x_i^{k+1}, \forall i, j$

$x_1^k(org) = 0, x_1^{k+1}(org) = 1, x_2^k(org) = 1, x_2^{k+1}(org) = 0$ (orijinal ve en iyi atama olarak kaydedilecek.)

$i = 3; i \leq n; i ++$ (bütün parçalar için)

$l = 2; l \leq i - 1; l ++$ (mevcut çizelgedeki bütün aralıklar için)

$h = i; h \geq l + 1; h --$ (yerleştirilecek aralıktan sonra gelen parçaları bir parça ileriye kaydır)

$$x_h^k = x_{h-1}^k, x_h^{k+1} = x_{h-1}^{k+1}.$$

bitiş döngü {h}

$x_l^k = 1, x_l^{k+1} = 0$, verilen atama ile yayılma zamanı hesapla ve C_{max}^k olarak kaydet.

$x_l^k = 0, x_l^{k+1} = 1$, verilen atama ile yayılma zamanı hesapla ve C_{max}^{k+1} olarak kaydet.

$Minmkspn(l) = \min\{C_{max}^k, C_{max}^{k+1}\}$, en iyi atamayı belirle.

$x_l^k = x_l^k(org), x_l^{k+1} = x_l^{k+1}(org), \forall l$, (orijinal atamaya geri dön).

bitiş döngü {1}

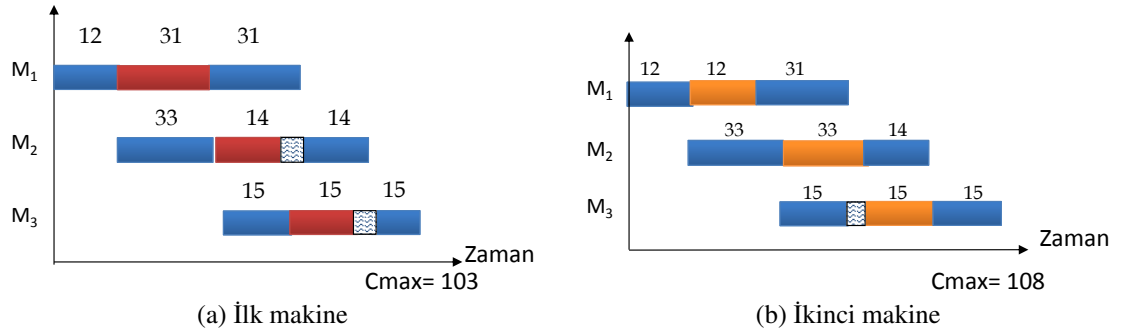
$MinC_i^m = \min_l\{Minmkspn(l)\}$, en iyi ve orijinal atamaları güncelle.

bitiş döngü {i}

Algoritmanın işleyişini bir örnekle açıklayalım.

Örnek 8 Parça sayısı 4, makine sayısı 3 olan sınırsız ara stok kapasitesine sahip bir örnek ele alalım. Esnek işlem yapan makineler 1'inci ve 2'nci makineler olsun. Makinelerin sabit işlem zamanları sırasıyla 12, 14 ve 15, esnek işlem zamanı 19 olsun ($n = 4, m = 3, k = 1, f^1 = 12, f^2 = 14, f^3 = 15, s = 19$).

Adım 1'e göre ilk parça ikinci makineye, ikinci parça da birinci makineye atanır. Adım 2'de anlatıldığı gibi üçüncü parçanın mevcut iki parça arasındaki aralıkta hem birinci hem de ikinci makineye atandığı durumlar ayrı ayrı ele alınır. Üçüncü parçanın birinci makineye atandığı durumdaki tamamlanma zamanı 103 (Şekil 6.1a), ikinci makineye atandığı durumdaki tamamlanma zamanı 108 (Şekil 6.1b) olarak hesaplanır. Üçüncü parçanın ataması, tamamlanma zamanı daha küçük olarak hesaplanan ilk makineye sabitlenir ve 3 parçalı bu sistemin tamamlanma zamanı 103 olarak bulunur.

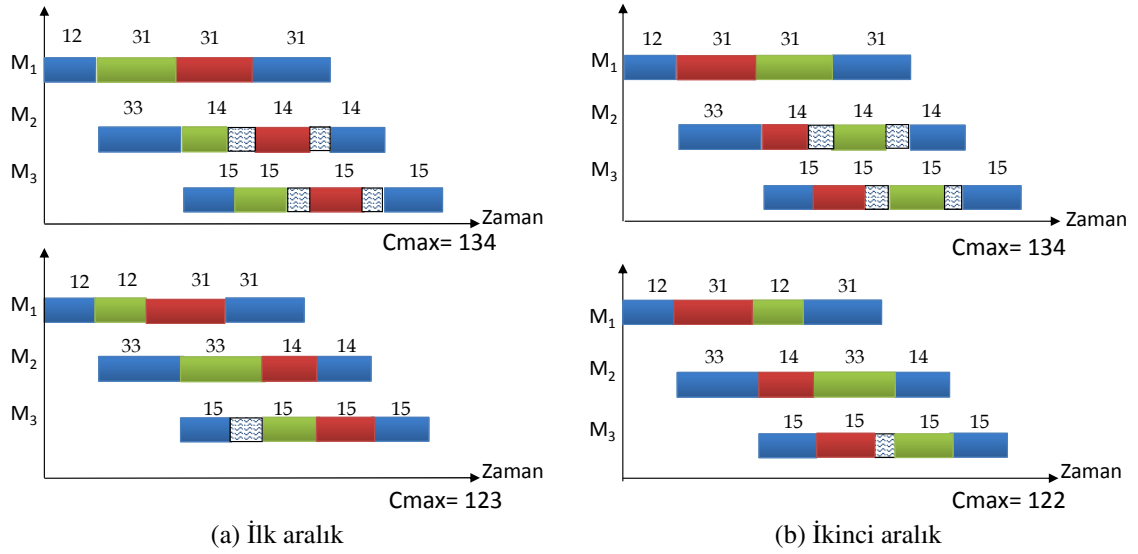


Şekil 6.1: İki parçadan üçüncü parçaya geçişte olası atama alternatifleri

Üçüncü parçadan dördüncü parçaya geçişte, Şekil 6.2’de görüldüğü gibi, yeni parça önce birinci ve ikinci parça arasına yerleştirilip birinci ve ikinci makineye ayrı ayrı atanır. Şekil 6.2a’da görüldüğü gibi, Adım 2’ye göre tamamlanma zamanları 134 ve 123 olarak hesaplanır. Sonra yeni parçanın ikinci ve üçüncü parça arasında ataması yapılır. Tamamlanma zamanları Şekil 6.2b’de görüldüğü gibi 134 ve 122 olarak bulunur. Hesaplanan bu dört tamamlanma zamanı kıyaslanır. Yeni parça en küçük tamamlanma zamanını veren, ikinci ile üçüncü parça arasına yerleştirilir ve bu parçanın esnek işlemi ikinci makineye atanır ve sabitlenir.

Böylece 1’inci parça 2’nci makineye, 2’nci parça 1’inci makineye, 3’üncü parça 2’nci makineye ve 4’üncü parça 1’inci makineye atanmış olur ve 4 parçalı bu örnek için tamamlanma zamanı 122 olarak bulunur. Bu problem için bulunan yayılma zamanı, optimal yayılma zamanına eşittir.

Sıfır ara stok problemi ve sınırsız ara stok problemi için geliştirilen algoritmalar aynı yaklaşımı kullanmaktadır. Bu iki algoritma arasındaki tek fark, yayılma zamanı hesabıdır. Problemlerin özelliğinden dolayı başlangıç ve tamamlanma zamanları farklı şekillerde hesaplanmaktadır, bu durum da yayılma zamanlarının farklı olması sonucunu doğurmaktadır.



Şekil 6.2: Üçüncü parçadan dördüncü parçaya geçişte olası atama alternatifleri

6.2 Deneysel Çalışma

Geliştirilen algoritmanın değerlendirilmesi amacıyla bir test prosedürü hazırlanmıştır. Problem için önemli parametreler belirlenmiş ve çeşitli değişiklikler yapılarak deneyler yapılmıştır.

Test parametreleri parça sayısı (n), makine sayısı (m), esnek işlem yapan makineler (k), işlem sürelerinin türetildiği taban değeri ($taban$), sabit işlem sürelerinin türetildiği aralık ($aralık - f$), esnek işlem sürelerinin türetildiği aralık ($aralık - s$) olarak belirlenmiştir. Parça ve makine sayılarının artması problem boyutunu artırmaktadır. Esnek makinelerin öncesinde makine olup olmadığı ve bu makine ya da makinelerin işlem süreleri problemin boyutunu artırıp azaltabilir, böylece çözüm süresini etkileyebilir. Esnek ve sabit işlem sürelerinin türetildiği aralıklar, işlem süresi değerlerini oluştururlar. İşlem sürelerine göre problem kolayca çözümlenebilir ya da atama varyasyonları artarak karmaşıklığı artırabilir. Sonuç olarak tüm bu parametreler, problemin boyutunu ve karmaşıklığını etkileyen, çözüm süresini ve kalitesini etkileyebilecek önemli parametrelerdir.

Belirlenen 6 parametre için, esnek her iki makine göz önüne alınarak 10'ar deneme yapılmak üzere sınırsız ve sıfır ara stok problemlerinin her biri için 640'ar adet deneme yapılmıştır. Algoritma Visual C/C++ 2008'de kodlanmış, denemeleri Intel core I5 3.2

GHz 4GB 64 bit bir bilgisayarda yapılmıştır. Yine aynı bilgisayarda matematiksel model GAMS CPLEX 12.2 kullanılarak çözdürülmüş ve her bir deneme 900sn ile sınırlı tutulmuştur. Yapılan denemeler, ele alınan çeşitli kriterler göz önünde tutularak karşılaştırılmıştır.

	Düşük	Yüksek
n	25	50
m	3	15
k	1	2 veya 7
$taban$	10	50
$aralık-f$	10	100
$aralık-s$	10	100

Çizelge 6.1: Test parametreleri

Test kriterlerinin belirlenmesinde her bir parametre için Çizelge 6.1’de görüldüğü gibi düşük ve yüksek olmak üzere ikişer değer belirlenmiştir.

Esnek makinelerin değerlendirilmesi için sistemdeki ilk iki makinenin ve ortada yer alan makinelerin seçilmesi düşünülmüştür. Esnek makinelerin öncesinde herhangi bir makine olmaması durumu esnek makinelerin ilk iki makine olarak alınması ile incelenecektir. Esnek makinelerin öncesinde ve sonrasında sabit makineler olabilir. Bu sabit makinelerin esnek makinelere etkisinin incelenmesi için ortada yer alan makineler esnek makine olarak alınabilir. Bu durum 3-makinelili sistemde 1’inci ve 2’nci; 15-makinelili sistemde 1’inci ve 7’nci makinelerin esnek makineler olarak alınmasına neden olmuştur. Ayrıca Önteorem 1’de anlatılmış olan ters problem özelliğinden dolayı son makinelerin esnek makineler olarak seçildiği durumların ayrıca incelenmesine gerek yoktur.

Sabit işlem zamanları, daha geniş bir aralıkta deneme yapılabilmesi için $f^j = DD[taban; taban + aralık-f]$ formülüne göre hesaplanmıştır. Burada $DD[a, b]$, a ve b arasında düzgün dağılımı ifade etmektedir. Parametre değerleri yerine konulduğunda sabit işlem zamanlarının $[10, 20]$, $[10, 110]$, $[50, 60]$, $[50, 150]$ aralıklarında düzgün dağılıma göre hesaplandıkları görülmektedir. Verilerin elde edilmesinde düzgün dağılımın kullanılmasının nedeni Hall ve Posner [31] ile Rardin ve Uzsoy [30]’un çizelgeleme problemlerinin çözümü için geliştirilmiş sezgisel yöntemlerin testi için, düzgün dağılımla türetilmiş verilerin kullanılmasının uygun olduğunu göstermiş olmalarıdır.

Önteorem 3’de gösterilen $|f^k - f^{k+1}| \geq s$ eşitsizliğinin geçerli olduğu durumlarda atama bilindiği için esnek işlem sürelerinin türetildiği aralığın $s = DD[|f^k - f^{k+1}|, |f^k - f^{k+1}| + aralık-s]$ olması uygun görülmüştür. Böylece esnek işlem sürelerinin türetildiği aralıklar, esnek makinelerin sabit işlem süreleri farkının mutlak değeri ile bu değere 10 veya 100 eklenmesi ile hesaplanmış olacaktır. Algoritmanın çözüm kalitesinin test edilmesi için yüzde hata değeri kullanılacaktır. Azami hata oranı denemelerin en yüksek hata oranını verirken; ortalama yüzde hata hesaplaması her bir denemede elde edilen hataların aritmetik ortalamasıdır. Yüzde hata oranı, her bir deneme için (Algoritma ile hesaplanan yayılma zamanı - Optimal yayılma zamanı / Optimal yayılma zamanı) formülü ile hesaplanmaktadır.

6.3 Sonuçlar ve Değerlendirme

Bu bölümde sınırsız veya sıfır ara stoklu sistemler için yapılan testler ve sonuçları ayrı ayrı sunulmuştur.

6.3.1 Sınırsız ara stok problemi

Çizelge 6.2’de görüldüğü gibi belirlenen parametreler dahilinde yapılan toplam 640 deneme sonucunda, sadece 17 örnekte optimal sonuca ulaşamamıştır. Bu örneklerdeki azami yüzde hata oranı % 0,49, ortalama yüzde hata ise % 0,086’dur. Genel çizelgenin haricindeki çizelgeler, denemelerin belirli kriterlere göre gruplandırılması ile oluşturulmuştur.

25 ve 50 parçanın 3 ve 15 makinelik örneklerinin her biri için 160 adet deneme yapılmıştır. Bu denemeler sonucunda Çizelge 6.2’de görüldüğü gibi 25 ve 50 parça için 3 makineden 15 makineye geçişte optimal olmayan çözüm sayısı ve ortalama yüzde hata miktarlarında artış gözlemlenmiştir. Algoritmanın çözüm hızında da 3 makineden 15 makineye geçişte bir artış gözlenmiş ancak GAMS çözüm süresi ile karşılaştırıldığında oldukça düşük oranlarda olduğu görülmüştür. GAMS’in çözümü için 900 sn’lik süre limitini 25 parçalı durumlarda geçen örnek bulunmazken 50 parçalık durumlarda süre limitini geçen örnek sayısı toplam 123 adettir. Bu durumda da 3 makineden 15 makineye geçişte bir artış gözlenmiştir.

Parça sayısı	Makine sayısı	Deneme sayısı	Optimal olmayan çözüm sayısı	Ortalama % hata	Azami % hata	CPU algoritma (sn)	CPU GAMS (sn)	Süre limitini geçen deneme sayısı (GAMS)
25	3	160	3	0,041	0,49	0,001	12,8	0
	15	160	9	0,086	0,44	0,004	47,3	0
50	3	160	1	0,007	0,11	0,005	373,7	49
	15	160	4	0,031	0,12	0,027	421,1	74

Çizelge 6.2: Sınırsız ara stok-genel

Esnek işlem sürelerinin türetildiği aralığa göre sonuçlar düzenlendiğinde, 25 ve 50 parçalık örneklerde esnek işlem sürelerinin türetildiği aralık 10'dan 100'e çıkarıldığında optimal olmayan çözüm sayılarında ve GAMS çözüm sürelerinde artış görülmüştür. Çizelge 6.3'de de görüldüğü gibi algoritmanın çözüm sürelerindeki artış oldukça düşüktür.

Parça sayısı	Makine sayısı	(s) Türetilen aralık	Deneme sayısı	Optimal olmayan çözüm sayısı	Ortalama % hata	Azami % hata	CPU algoritma (sn)	CPU GAMS (sn)	Süre limitini geçen deneme sayısı (GAMS)
25	3	10	80	1	0,015	0,12	0,001	4,4	0
		100	80	2	0,067	0,49	0,001	21,3	0
	15	10	80	0	0	0	0,004	22,1	0
		100	80	9	0,172	0,44	0,004	72,6	0
50	3	10	80	0	0	0	0,005	172,6	15
		100	80	1	0,014	0,11	0,005	574,8	34
	15	10	80	1	0,031	0,24	0,027	212,2	18
		100	80	3	0,039	0,12	0,027	630	56

Çizelge 6.3: Sınırsız ara stok-esnek işlem sürelerinin türetildiği aralığa göre

Yapılan denemeler, sabit işlemlerin türetildiği aralığa göre düzenlendiğinde Çizelge 6.4'de de görüldüğü gibi 25 parça için, 3 makinede aralık uzunluğu 10'dan 100'e çıkarıldığında; 50 parçada ise 100'den 10'a indirildiğinde hem ortalama yüzde hatalarda hem de optimal olmayan çözüm sayılarında artış görülmüştür. Azami yüzde hatalarda belirgin bir değişim görülmemesine rağmen, 100'den 10'a geçiş sırasında GAMS çözüm süresinde artış meydana gelmiştir.

Sonuç olarak sınırsız ara stok probleminde;

- Makine ve parça sayısı arttıkça,
- Sabit işlem sürelerinin türetildiği aralık daraldıkça,
- Esnek işlem sürelerinin türetildiği aralık genişledikçe,

Parça sayısı	Makine sayısı	(f^j) Türetilen aralık	Deneme sayısı	Optimal olmayan çözüm sayısı	Ortalama % hata	Azami % hata	CPU algoritma (sn)	CPU GAMS (sn)	Süre limitini geçen deneme sayısı (GAMS)
25	3	10	80	1	0,015	0,12	0,001	22,7	0
		100	80	2	0,068	0,49	0,001	2,9	0
	15	10	80	3	0,039	0,16	0,004	89,1	0
		100	80	6	0,133	0,44	0,004	5,5	0
50	3	10	80	1	0,014	0,11	0,005	484,3	34
		100	80	0	0	0	0,005	263,1	15
	15	10	80	4	0,062	0,24	0,03	603,6	53
		100	80	0	0	0	0,024	238,6	21

Çizelge 6.4: Sınırsız ara stok-sabit işlem sürelerinin türetildiği aralığa göre

problem zorlaşır ve çözüm süresi artar. Örneğin sabit işlem sürelerinin türetildiği aralık daraldıkça, yani sabit işlem süreleri birbirine yakın değerler aldıkça, esnek işlem atamasında en uygun sonucu bulmak için denemesi gereken alternatif sayısı artar. Böylece çözüm süresi artmış olur. Ancak algoritmanın çözüm kalitesinde ve süresinde önemli bir değişiklik olmamıştır.

6.3.2 Sıfır ara stok problemi

Sınırsız ara stok problemi çözüm algoritması sıfır ara stok problemi çözümü için yeniden düzenlenmiştir. Sıfır ara stok probleminin çözüm algoritması denemeleri için, sınırsız ara stok problemi çözümünde geliştirilmiş algoritmanın düzenlenmesi ile yeni çözüm yöntemi belirlenmiştir. Bu algoritma, problemin özelliği gereği sınırsız ara stok probleminin matematiksel modeline bazı kısıtların eklenmesi ile oluşan, yeni matematiksel modelin ürettiği optimal sonuçlar ile karşılaştırılmıştır. Belirlenen kriterler, kullanılan bilgisayar özellikleri gibi para-metreler sıfır ara stok denemeleri için de aynıdır. Çizelge 6.5’de de görüldüğü gibi, yapılan 640 denemenin sadece 2 tanesinde optimal sonuca ulaşılmamış, en yüksek hata oranı % 0,09 olarak hesaplanmıştır. Algoritmanın çözüm süresi GAMS’in çözüm süresine göre oldukça düşüktür. 640 denemenin toplam 88 tanesi süre limitini geçmiştir.

Esnek işlem sürelerinin türetildiği aralığa göre ise, Çizelge 6.6’da görüldüğü gibi GAMS’in çözüm süresinin aralık 100’den 10’a indirildiğinde arttığı gözlemlenirken, algoritmanın çözüm süresindeki artış % 0,001 düzeyinde kalmıştır.

Çizelge 6.7’de sabit işlem sürelerinin türetildiği aralık 100’den 10’a indirildiğinde

Parça sayısı	Makine sayısı	Deneme sayısı	Optimal olmayan çözüm sayısı	Ortalama % hata	Azami % hata	CPU algoritma (sn)	CPU GAMS (sn)	Süre limitini geçen deneme sayısı (GAMS)
25	3	160	1	0,004	0,06	0,002	0,92	0
	15	160	0	0	0	0,006	1,39	0
50	3	160	1	0,006	0,09	0,01	278,3	44
	15	160	0	0	0	0,037	271,4	44

Çizelge 6.5: Sıfır ara stok-genel

Parça sayısı	Makine sayısı	(s) Türetilen aralık	Deneme sayısı	Optimal olmayan çözüm sayısı	Ortalama % hata	Azami % hata	CPU algoritma (sn)	CPU GAMS (sn)	Süre limitini geçen deneme sayısı (GAMS)
25	3	10	80	0	0	0	0,002	1,16	0
		100	80	1	0,007	0,06	0,001	0,69	0
	15	10	80	0	0	0	0,007	1,45	0
		100	80	0	0	0	0,006	1,4	0
50	3	10	80	1	0,012	0,09	0,01	297,16	25
		100	80	0	0	0	0,011	260,41	19
	15	10	80	0	0	0	0,023	331,83	20
		100	80	0	0	0	0,024	280,11	24

Çizelge 6.6: Sıfır ara stok-esnek işlem sürelerinin türetildiği aralığa göre

GAMS'in çözüm süresinde artış görülmektedir.

Parça sayısı	Makine sayısı	(f ^j) Türetilen aralık	Deneme sayısı	Optimal olmayan çözüm sayısı	Ortalama % hata	Azami % hata	CPU algoritma (sn)	CPU GAMS (sn)	Süre limitini geçen deneme sayısı (GAMS)
25	3	10	80	0	0	0	0,002	1,07	0
		100	80	1	0,007	0,06	0,001	0,78	0
	15	10	80	0	0	0	0,007	1,43	0
		100	80	0	0	0	0,006	1,34	0
50	3	10	80	1	0,012	0,09	0,011	283,8	23
		100	80	0	0	0	0,01	273,76	21
	15	10	80	0	0	0	0,039	306,88	24
		100	80	0	0	0	0,034	235,96	20

Çizelge 6.7: Sıfır ara stok-sabit işlem sürelerinin türetildiği aralığa göre

Sıfır ara stok probleminde,

- Parça sayısı arttıkça,
- Sabit işlem ve esnek işlem sürelerinin türetildiği aralık daraldıkça,

problem zorlaşır ve çözüm süresi artar. Ancak sınırsız ara stok probleminde olduğu gibi algoritmanın sıfır ara stok problemi çözümünde de çözüm süresi ve kalitesinde bir kötüleşme meydana gelmemektedir.

Tüm bu sonuçlarla açıkça görülmektedir ki, geliştirilen sezgisel çözüm yöntemi n iş m makine akış tipi üretim sistemi problemi için hem sınırsız hem de sıfır ara stok kapasitesi durumlarında, oldukça kısa zamanda kaliteli sonuçlar vermektedir. Ayrıca makine sayısı ve parça sayısı ile işlem sürelerinin artması verilen algoritmanın çözüm süresinde ciddi bir artışa neden olmamıştır.

7. Sonuç ve Değerlendirme

Bu çalışmada n parça ve m makineli akış tipi üretim sistemi çizelgeleme problemi ele alınmıştır. k ve $k + 1$ 'inci makineler esnek makinelerdir ve diğer makinelerden farklı olarak belirli bir işlemi her iki makinenin de yapabilme kabiliyeti vardır. Makineler arası stok alanı sınırsız veya sıfır olabilir. Problemin amacı, yayılma zamanını enküçükleyecek şekilde her bir parça için esnek işlemlerin bu iki makineden hangisinde yapılacağına karar vermektir.

Problem detaylı bir şekilde tanımlandıktan sonra, 0-1 karma tamsayı program olarak modellenmiştir. Ancak özellikle makine ve parça sayısının çok olduğu durumlarda optimizasyon modelinin çözüm süresinin çok uzun olduğu, bazı örnekler için 24 saatlik bir sürede dahi çözüm vermediği, görülmüştür. Optimal çözümlerin özellikleri belirlenerek parametrelere bağlı olarak bazı problemlerin kolaylıkla çözüldüğü, bazılarının ise makine sayısı azaltılarak problem boyutunun küçültüldüğü belirlenmiştir. Hem sıfır ara stok hem de sınırsız ara stok için optimal sonuç veren dinamik programlama algoritmaları geliştirilmiştir. Bu algoritmalar öncelikle 3 ve 4 makineli sistemlerin özel durumları için geliştirilmiştir. Sonrasında ise m makine için verilen genel bir problemin 3 ve 4 makineli sistem durumlarından birisine indirgenebileceği gösterilmiştir. Bu algoritma problem parametrelerinin büyüklüğüne bağlı olarak yapay polinom zamanda çalışmaktadır. Ancak yine makine ve parça sayısının artması ile çözüm zamanı çok fazla artmaktadır. Bu yüzden kısa sürede çözüm verecek bir sezgisel yöntem geliştirilmiştir. Polinom zamanlı çözüm veren bu yöntem optimal sonucu garanti etmemektedir. Sezgisel yöntem sonuçlarının değerlendirilmesi için, kapsamlı bir deneysel çalışma yürütülmüştür. Problemi zorlaştıracığı düşünülen çeşitli kriterler belirlenmiş, sınırsız veya sıfır ara stok problemleri için ayrı ayrı denemeler yapılmıştır. Bilgisayarda kodlanan sezgisel yöntem, GAMS CPLEX 12.2 kullanılarak çözdürülen optimizasyon modeli sonuçları

ile karşılaştırılmıştır. Her bir ara stok seviyesi için 640'ar adet deneme yapılmış ve bu denemeler çeşitli başlıklar altında incelenerek yorumlanmıştır.

Sonuç olarak, geliştirilen sezgisel algoritmanın n parça ve m makineli sistemden oluşan esnek akış tipi çizelgeleme problemi için hem çözüm kalitesi hem de çözüm süresi açısından oldukça iyi sonuçlar verdiği görülmüştür. Problem verilerinin uygun olduğu durumlarda optimizasyon modeli ve dinamik programlama yaklaşımı optimal sonuçlar vermektedir. Ancak bu yöntemler özellikle makine ve parça sayısının büyük olduğu durumlarda kısa sürede çözüm vermeye elverişli değildir. Böyle bir problemle karşılaşıldığında sezgisel çözüm yöntemi kullanılmalıdır. Bu şekilde bir yaklaşımla, literatürde daha önce hiç çalışma yapılmamış olan bu alandaki boşluk doldurulmuştur. Ayrıca, esas amaç olan ve endüstride sıkça karşılaşılan genel m makineli ve karmaşık esneklik ilişkileri olan sistemlerin çözümü için önemli sonuçlar ve ipuçları elde edilmiştir.

Bu çalışmanın devamı niteliğinde olabilecek çeşitli çalışmalar yapılabilir. Bunlardan biri, esnek makine çiftlerinin birbirinden bağımsız olabileceği durumlardır. Ancak m makineli sistemde iki makineden fazla esnek makine olması, atamaların bir önceki makine çiftinin atamasından etkilenmesine neden olacağı için sisteme karmaşıklık katacaktır. Bunun sonucunda da bu çalışmada geliştirilen yöntemler doğrudan kullanılmayacak, yeni probleme adapte edilmesi ve geliştirilmesi gerekecektir. m makineli sistemde ardışık iki makine çifti yerine ardışık üç makine çifti olması durumu da ele alınabilecek diğer bir problemdir. Bu durumda, 3 esnek makinede bir tek esnek operasyon yapılabileceği gibi, ilk ve ikinci ile ikinci ve üçüncü makineler ayrı ayrı birer esnek operasyon yapabilirler. Bu oluşumlarda her bir esnek işlemin hangi makineye atandığı, makine dizilimine göre önde ve arkada olan makineleri etkileyecektir. Her bir ardışık makine çifti arasında esnek bir işlemin olduğu durum ise bu problemin daha genel bir halidir. Olabilecek en genel durumda ise, esnek işlemlerin mutlaka ardışık makinelerde olması kısıtı kaldırılarak, işler için atanma kuralları getirilebilir. Her bir i işinin atanabileceği makinelerin kümesi belirlenir ve her parçanın her işleminin hangi makinede yapılacağına karar verilir. Ele alınabilecek diğer bir durum makineler arasında teknolojik farkların olduğu sistemlerdir. Örneğin makinelerden bir tanesi diğerlerine göre daha yeni ve özellikli olabilir. Bunun sonucunda esnek işlem süreleri farklı olabilir ve esnek işlem daha yeni makineye atandığında daha kısa sürede işlem tamamlanır, fakat diğerine atandığında daha uzun sürebilir. Üretilecek parçaların tek tip olması kısıtı kaldırılarak değişik tipte parça üreten sistemlerde de

benzer esnekliklerle karşılaşılabilmektedir. Ama, bu problemler karmaşıklığı fazla olan sistemlerdir. m makineli değişik tipte parça üreten esnekliğin olmadığı klasik sistemlerde yayılma zamanı enküçüklemesi NP-Zor'dur. 2 makineli farklı tip parça üreten sistemlerin ise zayıf NP-Zor olduğu Gupta vd. [21] tarafından ispatlanmıştır. Bu çalışmada ele alınan problemin güçlü NP-Zor olup olmadığı ise bilinmemektedir.

KAYNAKLAR

- [1] Johnson, S. M., Optimal Two and Three-Stage Production Schedules with Setup Times Included, *Naval Research Logistics Quarterly*, 1, 61-68, 1954.
- [2] Pinedo, M., *Scheduling Theory, Algorithms and Systems*, 3. baskı, Springer Science Business Media, Prentice Hall, 2008.
- [3] Tanaev, V.S., Sotskov, Y.N., Strusevich, V.A., *Scheduling Theory: Multistage Systems*, Kluwer Academic Publishers, Dordrecht, 1994.
- [4] Nawaz, M., Enscore, E., Ham, I., A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem, *Omega*, 11, 91-95, 1983.
- [5] Gupta, N.D.J., Stafford, E., Flow Shop Scheduling Research After Five Decades, *European Journal of Operational Research*, 169, 699-711, 2006.
- [6] Hejazi, R., Saghafian, S., Flowshop Scheduling Problems with Makespan Criterion: A Review, *International Journal of Production Research*, 43(14), 2895-2929, 2005.
- [7] Becker, C., Scholl, A., A Survey on Problems and Methods in Generalized Assembly Line Balancing, *European Journal of Operations Research*, 168, 694-715, 2006.
- [8] Scholl, A., Becker, C., State-of-The-Art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing, *European Journal of Operational Research*, 168, 666-693, 2006.
- [9] Sethi, A., Sethi, S., Flexibility in Manufacturing: A Survey, *The International Journal of Flexible Manufacturing Systems*, 2, 289-328, 1990.

- [10] Gerwin, D., Manufacturing Flexibility: A Strategic Perspective, *Management Science*, 39, 395-410, 1993.
- [11] Groote, X., The Flexibility of Production Process: A General Framework, *Management Science*, 40, 7, 933-945, 1994.
- [12] Beach, R., Muhlemann, A.P., Price, D.H.R., Paterson, A., Sharp, J.A., A Review of Manufacturing Flexibility, *European Journal of Operational Research*, 122, 41-57, 2000.
- [13] Ruiz-Torres, A., Ho, J., Ablanedo-Rosas, H., Makespan and Workstation Utilization Minimization in a Flowshop with Operations Flexibility, *Omega*, 39, 273-282, 2011.
- [14] 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.
- [15] C., Tang, B., Tomlin. The power of flexibility for mitigating supply chain risks, *International Journal of Production Economics*, 116(1), 12-27, 2008.
- [16] Jordan, W., Graves, C., Principles on the Benefits of Manufacturing Process Flexibility, *Management Science*, 41(4), 577-594, 1995.
- [17] Daniels, R., Mazzola, J., Dailun, S., Flow Shop Scheduling with Partial Resource Flexibility, *Management Science*, 50(5), 658-669, 2004.
- [18] G., Nomden, D.J., van der Zee, Virtual Cellular Manufacturing: Configuring Routing Flexibility. *International Journal of Production Economics*, 112(1), 439-451, 2008.
- [19] Daniels, R., Mazzola, J., Flow Shop Scheduling with Resource Flexibility, *Operations Research*, 42(3), 504-522, 1994.
- [20] Burdett, R.L., Kozan, E., Sequencing and Scheduling in Flowshops with Task Redistribution, *Journal of the Operational Research Society*, 52, 1379-1389, 2001.
- [21] 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.

- [22] Askin, R., Chen, J., Dynamic Task Assingment for Throughput Maximization with Worksharing, *European Journal of Operational Research*, 168, 853-869, 2006.
- [23] Anuar, R., Bukchin, Y., Design and Operation of Dynamic Assembly Lines Using Work Sharing, *International Journal of Production Research*, 44, 4043-4065, 2006.
- [24] Crama, Y., Gultekin, H., Throughput Optimization in Two Machine Flowshops with Flexible Operations, *Journal of Scheduling*, 13(3), 227-243, 2010.
- [25] Suarez, F., Cusumano, M. ve Fine, C., An Empirical Study of Manufacturing Flexibility in Printed Circuit Board Assembly, *Operations Research*, 44, 223-240, 1996.
- [26] Gultekin, H., Akturk, S., Karasan, O. E., Cyclic Scheduling of a 2-Machine Robotic Cell with Tooling Constraints, *European Journal of Operational Research*, 174, 777-796, 2006.
- [27] Néron, E., Tercinet, F., Eloundou, A., Scheduling Identical Jobs on Uniform and Multi-Purpose Machines, *Inform's Manufacturing and Logistics Cluster*, San Antonio, Texas, USA, Kasım 2000.
- [28] Lee, C.Y., Lei, L., Pinedo, M., Current Trends in Deterministic Scheduling, *Annals of Operations Research*, 70, 1-41, 1997.
- [29] Muth, E., Reversibility Property of Production Lines, *Management Science*, 25,152-158, 1979.
- [30] Rardin, R., Uzsoy, R., Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial, *Journal of Heuristics*, 7, 261-304, 2001.
- [31] Hall, N., Posner, M., Generating Experimental Data for Computational Testing with Machine Scheduling Applications, *Operations Research*, 49, 854-865, 2001.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı : DEMİRTAŞ, Hatice Hande
Uyruğu : T.C.
Doğum tarihi ve yeri : 05.08.1987 Niksar - TOKAT
Medeni hali : Bekar
Telefon : 0535 256 6756
e-mail : hdemirtas@etu.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Y. Lisans	TOBB Ekonomi ve Teknoloji Üniversitesi Endüstri Mühendisliği	2011
Lisans	Gazi Üniversitesi Endüstri Mühendisliği	2009

Konum

Yıl	Yer	Görev
2009-2011	TOBB Ekonomi ve Teknoloji Üniversitesi	Burslu Yüksek Lisans Öğrencisi

Yabancı Dil

İngilizce

Yayınlar

Demirtaş H.H., Gültekin H., m-Makinelı Esnek Operasyonlu Akış Tipi Sistemlerde Çizelgeleme, Yöneylem Araştırması ve Endüstri Mühendisliği 31. Ulusal Kongresi (YAEM), Sakarya, Temmuz 2011.