

**KISITLANDIRILMIŞ TESLİM TARİHLİ
VE SIRAYA-BAĞIMLI HAZIRLIK SÜRELİ
TEK MAKİNE ÇİZELGELEME PROBLEMLERİNDE
ERKENLİK VE GEÇLİK TOPLAMININ EN KÜÇÜKLENMESİ**

Müge Hanım ÖZDEMİR

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**OCAK 2007
ANKARA**

Müge Hanım ÖZDEMİR tarafından hazırlanan KISITLANDIRILMIŞ TESLİM TARİHLİ VE SIRAYA-BAĞIMLI HAZIRLIK SÜRELİ TEK MAKİNE ÇİZELGELEME PROBLEMLERİNDE ERKENLİK VE GEÇLİK TOPLAMININ EN KÜÇÜKLENMESİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Ertan GÜNER
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği ile Endüstri Mühendisliği Anabilim Dalında Yüksek lisans olarak kabul edilmiştir.

Başkan : Prof. Dr. Zülal GÜNGÖR

Üye : Doç. Dr. Ertan GÜNER

Üye : Doç. Dr. M. Ali AKÇAYOL

Tarih : 12/01/2007

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

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.

Müge Hanım ÖZDEMİR

**KISITLANDIRILMIŞ TESLİM TARİHLİ
VE SIRAYA-BAĞIMLI HAZIRLIK SÜRELİ
TEK MAKİNE ÇİZELGELEME PROBLEMLERİNDE
ERKENLİK VE GEÇLİK TOPLAMININ EN KÜÇÜKLENMESİ
(Yüksek Lisans Tezi)**

Müge Hanım ÖZDEMİR

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Ocak 2007

ÖZET

Bu tezde, sıraya-bağımlı hazırlık süreleri içeren tek makineli bir çizelgeleme problemi incelenmiştir. Problemde tüm işler ortak bir teslim tarihine sahip olup işlerin bu tarihten önce veya sonra tamamlanması arzu edilmez. Erken veya geç tamamlanmadan dolayı cezalar ortaya çıkar. Literatürde bu problem Erkenlik/Geçlik (E/G) problemi olarak bilinir. Burada temel amaç, erkenlik ve geçlik toplamını en küçükleme. Japonların ortaya koyduğu Tam Zamanında (TZ) üretim felsefesinin popülerliğinin artması ile E/G problemi de yaygınlaşmıştır.

Bu çalışmada, erkenlik ve geçlik eşit ağırlığa sahiptir ve teslim tarihi tüm işler için ortak ve kısıtlandırılmıştır. Çizelgeleme literatüründe genellikle makine hazırlık süreleri göz ardı edilir veya işlem sürelerinin içinde varsayılır. Bu çalışmada, makinenin hazırlık süresi işlem süresinden ayrı ve sıraya-bağımlı olarak ele alınmıştır. Hazırlık sürelerinin dikkate alınmasıyla problemin karmaşıklığı oldukça artmış ve NP-zor bir problem olmuştur.

Çalışmada, küçük boyutlu problemlerin en iyi çözümlerini elde etmek için bir Karışık Tamsayılı Programlama (KTP) modeli kullanılmıştır. Büyük boyutlu problemlerin en iyi çözümlerini KTP ile makul sürelerde bulmak mümkün

olmamıştır. Bu nedenle, daha önce kısıtlandırılmamış ortak teslim tarihli problem için geliştirilen düzenlenmiş en küçük işlem süreli işin ilk yerleştirilmesine dayanan *SAPT* sezgiseli kısıtlandırılmış duruma uyarlanarak büyük boyutlu problemler için çözümler elde edilmiştir. Bu çözümleri geliştirmek için *SAPT* sezgiseliyle bulunan çözüm Tabu Arama (TA) algoritmasının başlangıç çözümü olarak kullanılmıştır. Hem *SAPT* sezgiselinin ve hem de TA algoritmasının performansı, çözümlerinin en iyi çözümlerden sapması dikkate alınarak ölçülmüştür. *SAPT* sezgiseliyle kabul edilebilir çözümler bulunmakla birlikte, TA algoritması pek çok örnekte iyileştirme sağlamış ve iş sayısı 180'e kadar olan problemlerin çözümleri gerçekleştirilmiştir.

Bilim Kodu	: 906.1.141
Anahtar Kelimeler	: Erkenlik, geçlik, sıraya-bağımlı, kısıtlandırılmış, tek makine
Sayfa Adedi	: 109
Tez Yöneticisi	: Doç. Dr. Ertan GÜNER

**MINIMIZING THE TOTAL EARLINESS AND TARDINESS FOR SINGLE
MACHINE SCHEDULING PROBLEMS WITH RESTRICTED DUE DATE
AND SEQUENCE-DEPENDENT SETUP TIMES**

(M.Sc. Thesis)

Müge Hanım ÖZDEMİR

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
January 2007**

ABSTRACT

In this thesis, a single-machine scheduling problem including sequence-dependent setup times is discussed. All jobs have a common due date in this problem and it is not desired that the jobs complete before or after the due date. In the literature, this problem is known as the Earliness/Tardiness (E/T) problem. The main objective is minimizing the total amount of earliness and tardiness. The E/T problem became common by the popularity of the Just-In-Time (JIT) Japanese manufacturing philosophy increased.

In this research, earliness and tardiness have equal weights and the due date is common and restricted for all jobs. In the most scheduling literature, machine setup time was ignored or assumed to be part of the jobs' processing times. In this thesis, the machine setup is separated from the processing time and is considered to be sequence-dependent. The intricacy of the problem increases dramatically and the problem becomes NP-hard when sequence-dependent setup times are included. Therefore, optimal solutions cannot be obtained in a polynomial time.

A Mixed Integer Programming (MIP) is used to obtain optimal solutions for small sized problems. For large-sized problems, obtaining optimal solutions using exact methods is not feasible. Therefore, the “Shortest Adjusted

Processing Time First” (*SAPT*) heuristic developed before for unrestricted common due date of this problem is also adapted to restricted version. After, a Tabu Search (TS) algorithm is implemented to find better solutions using the best solution of the *SAPT* heuristic as an initial solution. The performance of the *SAPT* heuristic and the TS algorithm are measured by considering the deviations of their solutions from the optimal solutions. *SAPT* heuristic provided acceptable solutions; however, the TS algorithm outperformed the *SAPT* heuristic in the most cases and results for problems up to 180 jobs were found.

Science Code : 906.1.141

Key Words : Earliness, tardiness, sequence-dependent, restricted, single machine

Page Number : 109

Adviser : Assoc. Prof. Ertan GÜNER

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Hocam Doç.Dr. Ertan GÜNER'e, maddi ve manevi destekleriyle her zaman yanımda olan sevgili aileme, eşsiz yardımlarıyla ve sevgisiyle bana daima güç veren çok özel arkadaşım İsmail ŞENGÜZEL'e teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ	xii
ŞEKİLLERİN LİSTESİ	xiii
SİMGELER VE KISALTMALAR	xv
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	5
2.1. Ortak Teslim Tarihli Modeller	8
2.1.1. Ortak teslim tarihinden toplam mutlak sapmanın en küçüklenmesi	8
2.1.2. Farklı erkenlik ve geçlik cezaları	12
2.1.3. İlave cezalar	12
2.1.4. Doğrusal olmayan cezalar	15
2.1.5. İşe-bağımlı erkenlik ve geçlik cezaları	17
2.1.6. Paralel makineli modeller	18
2.1.7. Teslim tarihi toleransları.....	21
2.2. Farklı Teslim Tarihli Modeller	22
2.2.1. Boş zamanların yer almadığı modeller.....	23
2.2.2. Boş zamanların yer aldığı modeller	24
2.2.3. Farklı özelliklerdeki diğer problem tipleri	25
2.3. Zorunlu Teslim Tarihli Modeller	26

Sayfa

2.4. Sıraya-Bağımlı Hazırlık Süreleri İçeren Modeller	28
3. HAZIRLIK SÜRELERİ İÇEREN E/G PROBLEMLERİNİN ANALİZİ	31
3.1. Hazırlık Sürelerinin Yarattığı Zorluklar	32
3.2. Teoremler ve Özellikler	33
3.3. Hazırlık ve İşlem Süreleri için Matematiksel Eşitlik	35
4. GELİŞTİRİLEN ÇÖZÜM YÖNTEMLERİ	37
4.1. KTP Kullanarak E/G Problemleri için En İyi Çözümler Bulma	37
4.1.1. Hazırlık süreleri üçgensel olan E/G problemi için bir KTP	37
4.1.2. Modelin kurulumu.....	39
4.1.3. Veri üretimi	40
4.2. E/G Problemi için Yeni Bir Sezgisel Algoritma	43
4.3. Tabu Arama Algoritması	57
4.3.1. Basit bir TA algoritması	58
4.3.2. E/G çizelgeleme probleminde kullanılan TA algoritmaları	61
4.3.3. TA kullanarak çizelgeleme	63
4.3.4. Önerilen TA algoritması	65
5. DENEYSEL BULGULAR VE TARTIŞMA	68
5.1. <i>SAPT</i> Sezgiseli İçin İşlemsel Deney	68
5.1.1. Deneysel tasarım	68
5.1.2. İstatistiksel analiz	70
5.2. TA Algoritması İçin İşlemsel Deney	76
5.2.1. TA ile elde edilen sonuçlar ve sonuçların analizi	76
5.2.2. İstatistiksel analiz	77

Sayfa

5.3. TA Algoritması ile Sağlanan İyileştirme	84
6. KULLANICIYA YÖNELİK ARAYÜZ OLUŞTURMA	86
6.1. Arayüz Hakkında Kısa Bilgi	86
6.2. Arayüzü Oluşturan Sayfalar	86
6.3. Arayüzde Kullanılan Uyarı/Hata Mesajları ve İpucu Metinleri	91
7. SONUÇ VE ÖNERİLER	92
7.1. Çalışmanın Katkıları	92
7.2. Sonuçlar	93
7.3. Gelecek Çalışmalar için Öneriler	94
KAYNAKLAR	95
EKLER	103
EK-1 AMPL diliyle yazılan model	104
EK-2 KTP ile bulunan en iyi çözümler	105
ÖZGEÇMİŞ	109

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Birincil hedef ve ikincil hedefler.....	17
Çizelge 2.2. Tek-makinelı E/G problemi için literatür özeti.....	30
Çizelge 3.1. Sıraya-bağımlı hazırlık süreleri örneđi	31
Çizelge 3.2. Örnek 3.1 için işlem ve hazırlık süreleri.....	35
Çizelge 3.3. Örnek 3.1 için düzenlenmiş işlem süreleri	35
Çizelge 4.1. Örnek 4.1 için veriler.....	41
Çizelge 4.2. Örnek 4.1 için en iyi çözüm.....	42
Çizelge 4.3. En iyi çözümlerin özeti.....	42
Çizelge 4.4. Örnek 4.2 için işlem ve düzenlenmiş işlem süreleri.....	49
Çizelge 4.5. E/G problemini çözmek için kullanılan farklı TA metotları	63
Çizelge 5.1. <i>SAPT</i> deneylerindeki faktörler ve seviyeleri	69
Çizelge 5.2. <i>SAPT</i> sezgiseli için deney sonuçları.....	70
Çizelge 5.3. <i>SAPT</i> deneyleri için varyans analizi.....	71
Çizelge 5.4. TA algoritması için deney sonuçları - 1	76
Çizelge 5.5. Farklı problem büyüklüklerinin TA ile çözüm süreleri.....	77
Çizelge 5.6. TA algoritmasının 1. deneyi için varyans analizi	78
Çizelge 5.7. TA algoritmasının 2. deneyindeki faktörler ve seviyeleri	83
Çizelge 5.8. TA algoritmasının 2. deneyi için varyans analizi	83
Çizelge 5.9. TA algoritması için deney sonuçları – 2.....	85

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. İncelenen problemin yapısı.....	3
Şekil 3.1. Teslim tarihinde tamamlanmayan bir işin gösterimi.....	34
Şekil 4.1. Üçgensel eşitsizlik.....	39
Şekil 4.2. İş sayısı ile CPU süresi arasındaki ilişki	43
Şekil 4.3. TA'nın akış şeması.....	59
Şekil 5.1. <i>SAPT</i> için tam faktöriyel deney tasarımı	69
Şekil 5.2. <i>SAPT</i> sezgiselinde $n=5$ için h ile en iyiden % sapma arasındaki ilişki	72
Şekil 5.3. <i>SAPT</i> sezgiselinde $n=7$ için h ile en iyiden % sapma arasındaki ilişki	72
Şekil 5.4. <i>SAPT</i> sezgiselinde $n=9$ için h ile en iyiden % sapma arasındaki ilişki	73
Şekil 5.5. <i>SAPT</i> sezgiselinde $n=11$ için h ile en iyiden % sapma arasındaki ilişki ...	73
Şekil 5.6. <i>SAPT</i> sezgiselinde $h=0,4$ için n ile en iyiden % sapma arasındaki ilişki ..	74
Şekil 5.7. <i>SAPT</i> sezgiselinde $h=0,6$ için n ile en iyiden % sapma arasındaki ilişki ..	74
Şekil 5.8. <i>SAPT</i> sezgiselinde $h=0,8$ için n ile en iyiden % sapma arasındaki ilişki ..	75
Şekil 5.9. <i>SAPT</i> sezgiselinde tüm iş sayıları için h ile en iyiden % sapma arasındaki ilişki	75
Şekil 5.10. TA'da iş sayısı ile CPU süresi arasındaki ilişki.....	77
Şekil 5.11. TA'da $n=5$ için h ile en iyiden % sapma arasındaki ilişki.....	79
Şekil 5.12. TA'da $n=7$ için h ile en iyiden % sapma arasındaki ilişki.....	79
Şekil 5.13. TA'da $n=9$ için h ile en iyiden % sapma arasındaki ilişki.....	80
Şekil 5.14. TA'da $n=11$ için h ile en iyiden % sapma arasındaki ilişki.....	80
Şekil 5.15. TA'da $h=0,4$ için n ile en iyiden % sapma arasındaki ilişki.....	81
Şekil 5.16. TA'da $h=0,6$ için n ile en iyiden % sapma arasındaki ilişki.....	81

Şekil	Sayfa
Şekil 5.17. TA’da $h=0,8$ için n ile en iyiden % sapma arasındaki ilişki.....	82
Şekil 5.18. TA’da tüm iş sayıları için h ile en iyiden % sapma arasındaki ilişki.....	82
Şekil 6.1. İş çizelgeleme programı–ana sayfa	86
Şekil 6.2. İş çizelgeleme programı–örneklerden çalıştırma sayfası (açılış ekranı) ...	87
Şekil 6.3. İş çizelgeleme programı–örneklerden çalıştırma sayfası (çalıştırma sonrası)	88
Şekil 6.4. İş çizelgeleme programı–girilen değerlerle çalıştırma sayfası (açılış ekranı).....	89
Şekil 6.5. İş çizelgeleme programı–girilen değerlerle çalıştırma sayfası (çalıştırma sonrası)	90

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
n	İş sayısı
P_j	j işinin işlem süresi
C_j	j işinin tamamlanma zamanı
d_j	j işi için teslim tarihi
d	Tüm işler için ortak teslim tarihi
E_j	j işinin erkenliği
T_j	j işinin geçliği
α_j	j işi için birim erkenlik cezası
β_j	j işi için birim geçlik cezası
S_{ij}	Hazırlık süresi
AP_{ij}	Düzenlenmiş İşlem Süresi
A	Geç işler kümesi
B	Erken işler kümesi
a	A kümesinin eleman sayısı
b	B kümesinin eleman sayısı
C_A	Geç işlerin toplam cezası
C_B	Erken işlerin toplam cezası
Δ	Erken işler kümesinin uzunluğu
χ	Makine çalıştırma maliyeti
w_j	j işinin birim bekleme süresi maliyeti
C_s	Toplam beklenen E/G maliyeti
M	Büyük pozitif tamsayı
T	Tabu listesi
h	Teslim tarihini kısıtlandıran faktör
k	Örnek numarası

Kısaltmalar	Açıklama
E/G	Erkenlik/Geçlik
NP	Non-polinomial
TZ	Tam Zamanında
GSP	Gezgin Satıcı Problemi
KTP	Karışık Tamsayılı Programlama
D&S	Dal-ve-Sınır Algoritması
TA	Tabu Arama
API	Tüm Bitişik İkililerin Yer Değiştirmesi
GPI	Tüm İkililerin Yer Değiştirmesi
F-W	Floyd-Warshall
CPU	Merkezi İşlem Ünitesi
SAPT	Düzenlenmiş En Küçük İşlem Süreli
STTA	Sıra-Tabanlı Tabu Arama
EDD	En Erken Teslim Tarihi
ANOVA	Varyans Analizi

1. GİRİŞ

Tam zamanında (TZ) üretim felsefesinin yaygınlaşması ile müşteriler daha yüksek standartlara alışmaya başlamışlardır. Bu nedenle malları erken ya da geç almak istememektedirler. TZ dağıtım stratejisi üretici için bazı zorluklar yaratmaktadır. Bir TZ çizelgeleme yapısında, erken biten işler teslim tarihleri gelene kadar üreticinin elinde kalır. Bu durumda üretici, ürünün bozulmasından kaynaklanan maliyetler ile depolama veya sigorta gibi maliyetler ile karşılaşır. Diğer yandan, teslim tarihlerinden sonra tamamlanan işler müşteri tatminsizliği, satış kayıpları, sözleşme cezaları veya itibar kayıplarına neden olur.

Çizelgeleme, kıt kaynakların görevlere tahsis edilmesini ele alan, bir ya da daha fazla amacı gerçekleştirmeyi hedefleyen bir karar verme işlemidir. Makine çizelgelemede, kaynak olarak sınırlı kapasitedeki makineler veya işlemciler; görev olarak da ürünler, işler ya da kıt kaynak kullanımı gerektiren herhangi bir şey olarak tanımlanabilir. Her görev belirli bir öncelik seviyesine, en erken mümkün başlama zamanına ve bir teslim tarihine sahip olabilir. Farklı istekler doğrultusunda çizelgelemenin amaçları değişik biçimler alabilir.

TZ üretim kavramının benimsenmesi durumunda, ideal bir çizelge için tüm işlerin teslim tarihlerinde tamamlanması gerekir. İşlerin erken ya da geç tamamlanmasının cezalandırıldığı çizelgeleme modeli erkenlik/geçlik (E/G) çizelgeleme problemidir. Bu problemin amacı E/G cezalarının toplamını en küçükleyen iş çizelgesini belirlemektir. Ele alınan tek-makineli E/G problemi aşağıdaki varsayımlar altında değerlendirilmiştir:

- 1) Problem *deterministiktir*: Tüm iş parametreleri (işlem süreleri, hazırlık süreleri, E/G cezaları ve ortak teslim tarihi) kesin olarak bilinir.
- 2) Problem *statiktir*: Tüm işler sıfır zamanında hazırdır ve herhangi bir işin sistemden ayrılması beklenmez.
- 3) Teslim tarihi kısıtlandırılmış ve tüm işler için ortak yani $d_j = d$ olarak verilir.
- 4) Kesintiye izin verilmez: Bir makine bir işi işlemeye başladığında iş bitene kadar

ara vermez.

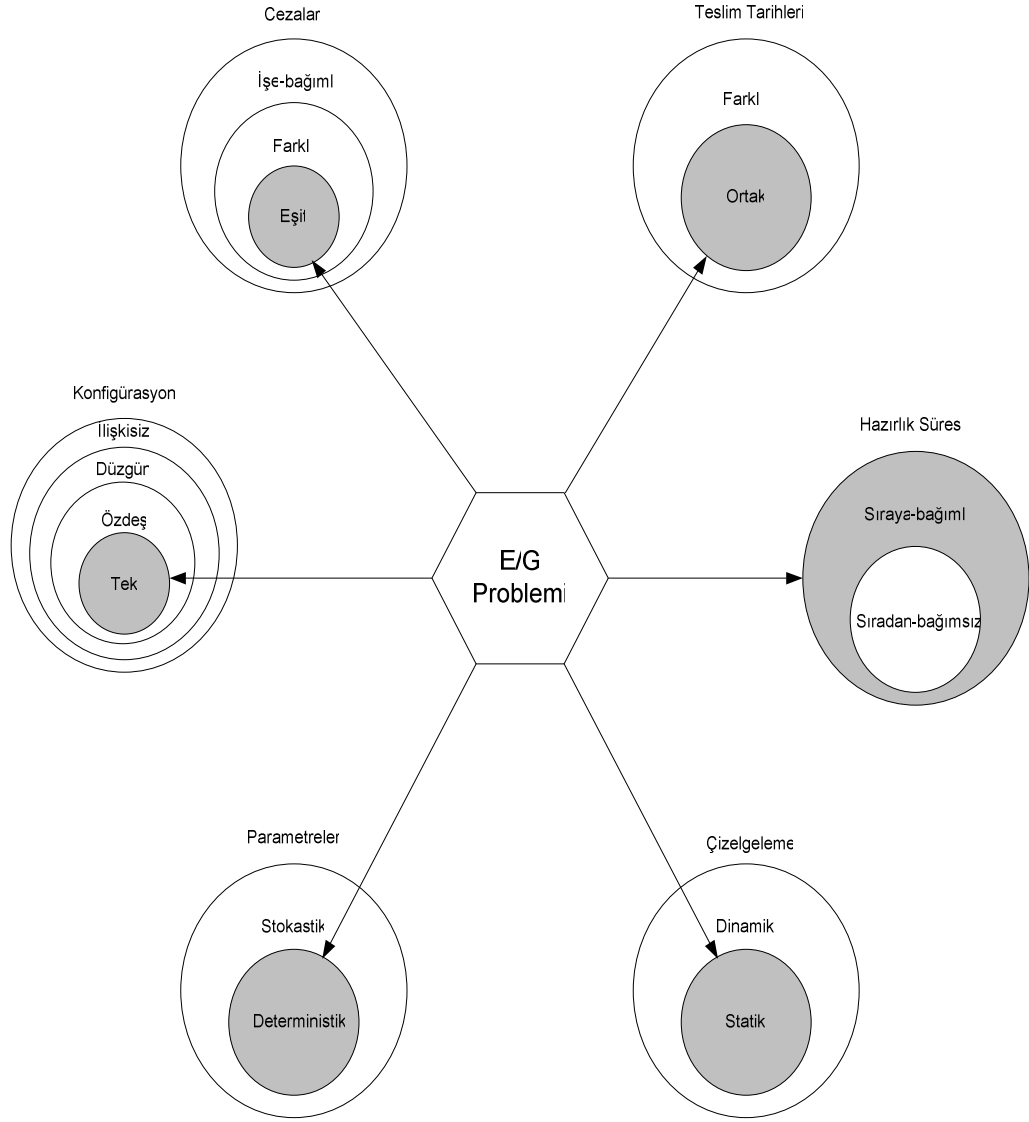
5) Erkenlik ve geçlik cezaları eşittir.

6) Hazırlık süreleri sıraya-bağımlıdır.

Gerçek endüstriyel uygulamalarda, sıradaki bir işin işlenmesi için makinenin hazırlanması gerekir. E/G literatürünün çoğunda makinenin, işleri işlemek üzere daima hazır olduğu ve hazırlığa ihtiyaç duymadığı varsayılır. Bazılarında ise hazırlık süresi işlem süresinin içinde yer alır. İkinci durum ilkinden çok farklı değildir çünkü her iki durumda da makinenin, işlerin sırasına bakılmaksızın yani sıradan-bağımsız olarak aynı hazırlık süresini gerektirdiği varsayılmıştır. Bazı durumlarda hazırlık süresi sıradaki pozisyonundan bağımsız olarak belli bir iş için aynı kalabilirken, çoğu zaman henüz tamamlanan işin tipi, işlenecek olan iş kadar makinenin hazırlık süresini etkileyebilir.

Sıraya-bağımlı hazırlık süreleri içeren tek-makineli E/G problemi çizelgeleme literatüründe dikkate alınmıştır. Ancak, bu incelemelerde problemin teslim tarihi tüm işler için ortaktır ve kısıtlandırılmamıştır. Bu çalışmada ise hem kısıtlandırılmış ortak teslim tarihi hem de sıraya-bağımlı hazırlık süreleri dikkate alınmıştır. Kısıtlandırılmış ortak teslim tarihinin probleme dâhil edilmesi problemin zorluğunu önemli ölçüde artırır. Diğer taraftan sıraya-bağımlı hazırlık süreleri de problemin çözümünü zorlaştıran diğer bir unsurdur. E/G problemlerinin sıradan-bağımsız ve kısıtlandırılmamış ortak teslim tarihli en basit durumu bile NP-zor bir problemdir. Bu çalışmada ele alınan E/G problemi belirttiğimiz özelliklerden dolayı NP-zor bir problemdir ve ele alınıp incelenmesi ile literatüre katkı sağlayacağı düşünülmektedir.

E/G problemi değişik şartlar altında incelenebilir. Örneğin, çok-makineli veya tek-makineli sistem için çözülebilir. Teslim tarihleri farklı veya ortak olabilir. E/G cezaları değişik şekilde; eşit, eşit olmayan, işe-bağımlı olabilir. Hazırlık süreleri sıradan-bağımsız ya da sıraya-bağımlı olabilir. Şekil 1.1’de bu şartlar ve çalışmaya konu olan problemin yapısı görülmektedir.



Şekil 1.1. İncelenen problemin yapısı

Çalışmanın ikinci bölümünde, E/G çizelgeleme problemlerini dikkate alan bir literatür araştırması yapılmıştır. Bu bölümde ortak, farklı ve zorunlu teslim tarihli modeller ile sıraya-bağımlı hazırlık süreleri içeren modeller incelenmiştir. Üçüncü bölümde ise sıraya-bağımlı hazırlık süreleri içeren problemlerin daha detaylı analizi yapılmıştır.

Geliştirilen çözüm yöntemleri dördüncü bölümde sunulmuştur. Bunlardan ilki en iyi çözümler bulan karışık tamsayılı programlamadır. E/G problemi, AMPL dili kullanılarak formüle edilmiş ve CPLEX 8.0 çözücüsü kullanılarak 11 işe kadar olan problemlerin en iyi çözümü bulunmuştur. Bu kısımda, araştırmanın diğer bölümlerinde de kullanılacak olan örnek setlerini oluşturan bir veri üretici de programlanmıştır. İkinci olarak, problemin kısıtlı olmayan versiyonu için geliştirilmiş olan *SAPT* sezgiseli kısıtlandırılmış duruma uyarlanmış ve en iyi/yaklaşık en iyi çözümler elde edilmiştir. Son olarak, *SAPT* sezgiseli ile bulunan en iyi çözüm, Tabu Arama (TA) algoritmasının başlangıç çözümü olarak alınmış ve birçok problemde iyileştirme sağlanmıştır.

Beşinci bölümde, en iyi çözümünden sapmalar temel alınarak *SAPT* sezgiseli ve TA algoritmasının performansları değerlendirilmiştir. Her ikisi için yapılan deney tasarımları sonucunda, iş sayısının ve teslim tarihini kısıtlandırma faktörünün deney üzerinde etkili olduğu ortaya çıkmıştır. TA algoritması için bir deney tasarımı daha yapılmış, bu sefer, tabu listesinin uzunluğu ve iterasyon sayısı faktörleri incelenmiştir. Her iki faktör de araştırılan seviyeleri için deney üzerinde anlamlı çıkmamıştır.

Kullanıcıya rahat ve zevkli kullanım sunması için arayüzler tasarlanmış ve bunlar altıncı bölümde anlatılmıştır. Arayüzlere gerekli veri girişleri yapılarak örnek problemler çalıştırılabileceği gibi kullanıcının kendi oluşturduğu problemleri çözme imkânı da vardır. Böylece sadece örnekler değil, gerçek yaşam problemleri de çözülebilmektedir.

Son olarak, yedinci bölümde, çalışmanın sonuçları, literatüre katkıları ve gelecek çalışmalar için öneriler sunulmuştur.

2. LİTERATÜR ARAŞTIRMASI

Bu bölümde öncelikle çalışmada kullanılan notasyonlar, çizelgeleme problemlerinin sınıflandırılması ve başlıca performans ölçütlerinden bahsedilmiştir. Daha sonraki kısımlarda, E/G problemlerini dikkate alan çizelgeleme problemleri incelenmiştir.

Notasyon ve Tanımlar

Bu kısımda çalışma boyunca kullanılan notasyon ve tanımlar listelenmiştir:

- n : İş sayısı
- p_j : j işinin işlem süresi
- C_j : j işinin tamamlanma zamanı
- d_j : j işi için teslim tarihi
- d : Tüm işler için ortak teslim tarihi
- h : Teslim tarihini kısıtlandırma faktörü
- E_j : j işinin erkenliği; $E_j = \max(0, d_j - C_j)$
- T_j : j işinin geçliği; $T_j = \max(0, C_j - d_j)$
- α_j : j işi için birim erkenlik cezası
- β_j : j işi için birim geçlik cezası
- S_{ij} : i işinden sonra gelen j işi için gereken hazırlık süresi
- AP_{ij} : Düzenlenmiş İşlem Süresi: i işinden sonra gelen j işinin hazırlık ve işlem süresi
- A : Geç işler kümesi; $A = \{1 \leq j \leq n \mid C_j > d\}$
- B : Erken işler kümesi; $B = \{1 \leq j \leq n \mid C_j \leq d\}$
- a : A kümesinin eleman sayısı; $a = |A|$
- b : B kümesinin eleman sayısı; $b = |B|$

Sınıflandırma

Makine Sayısına Dayalı Sınıflandırma ve Makine Yapılandırılmaları

Çizelgeleme literatürü makine yapılandırmalarını birkaç duruma göre sınıflandırır. Pinedo'ya [1] göre, çizelgeleme probleminde belirlenmiş bazı makine yapılandırmaları şöyledir:

Tek makine: Tüm mümkün makine yapılandırmalarının en basitidir ve daha karışık makine yapılandırmalarını karşılaştırmak için özel bir durumdur.

Paralel özdeş makineler: Paralel olarak m tane özdeş makine vardır. j işi tek bir işlem gerektirir ve m makineden birinde işlenebilir.

Paralel özdeş olmayan makineler: Paralel olarak m tane aynı işi yapabilen makine vardır. i makinesi, j işini v_{ij} hızıyla işleyebilir ve j işinin i makinesinde geçirdiği p_{ij} süresi p_j/v_{ij} 'ye eşittir.

Akış atölyesi: Seri olarak m tane makine vardır. Her iş m makinenin her birinde işlenmek zorundadır. Tüm işler aynı rotayı izlemelidir.

İş atölyesi: m makineli bir iş atölyesinde her iş önceden belirlenmiş rotasını izler. Akış atölyesiyle karşılaştırıldığında daha az kuyruk vardır.

Açık atölye: m tane makine vardır. Her iş m makineden birinde işlenmek zorundadır. İşlerin rotalarının belirlenmesine ilişkin bir kısıtlama yoktur.

İşlerin Hazır Bulunma Zamanlarına Göre Sınıflandırma

Eğer tüm işler sıfır zamanında veya çizelge yapılmadan önce işlem için hazır bulunuyorsa çizelgeleme problemi *statik* olarak adlandırılır. İşler zaman içinde hazır bulunuyorsa problem *dinamik* olur. Hem dinamik hem de statik problemler için işlem

süreleri, varış zamanları ve teslim tarihleri gibi iş parametreleri sabitse ve önceden biliniyorsa problem *deterministik*, aksi takdirde *stokastik* olarak adlandırılır.

İşlerin Teslim Tarihlerine Göre Sınıflandırma

Teslim tarihi çizelgeleme problemlerinde önemli bir rol oynar. Teslim tarihleri müşteriler tarafından belirlenebileceği gibi üretici tarafından da belirlenebilir. Tüm işlerin aynı teslim tarihine sahip olması durumunda *ortak* teslim tarihli problemler, aksi durumda ise *farklı* teslim tarihli problemler ortaya çıkar.

Hazırlık Sürelerine Göre Sınıflandırma

Gerçek hayat şartlarında, sıradaki bir işin işlenmesi için makinenin hazırlanması gerekir. Bazı durumlarda bu hazırlık süresi sıradaki pozisyonundan bağımsız olarak belli bir iş için aynı kalabilirken diğer durumlarda sıraya-bağımlı olabilir. Problemin karmaşıklığı hazırlık süresi sıraya-bağımlı olduğunda önemli şekilde artar.

Performans Ölçütleri

Çizelgeleme probleminde, her çizelgenin dikkate alınan karakteristikleri *performans ölçütleri* olarak adlandırılır. Performans ölçütleri genel olarak dört gruba ayrılır. Bunlar bazı problem örnekleriyle birlikte şöyledir:

Kaynakların verimli kullanımı: Sistemi terk eden son işin tamamlanma zamanı,

Taleplere hızlı cevap verme: Ortalama tamamlanma zamanı, ortalama akış zamanı, ortalama bekleme zamanı,

Belirli teslim tarihlerine yakınlık: Ortalama geçlik, en büyük geçlik, geç işlerin sayısı, erkenlik ve geçliğin toplamı,

Ürün stokundaki azalma: Ortalama erkenlik, en büyük erkenlik.

Günümüz şartlarında erken ve geç biten işler çeşitli sorunlar yaratır. Erken tamamlanan işlerin getirdiği bozulma, depolama, sigorta gibi maliyetler yanısıra

nihai ürünün stoku dolaylı olarak fırsat maliyeti taşıyan verimsiz bir yatırıma işaret eder [2].

Yukarıdaki sorunlar göz önüne alındığında TZ üretim kavramına olan ilgi artmış ve bu konuda önemli araştırmalar yapılmıştır. TZ üretim felsefesine sahip çizelgeleme problemlerinin en yaygın amaçlarından biri erkenlik ve geçlik cezalarının en küçüklenmesidir. Bu probleme erkenlik/geçlik çizelgeleme problemi denir. j işinin erkenlik ve geçliği olarak sırasıyla E_j ve T_j gösterimleri dikkate alınırsa genel bir E/G probleminde j işinin erkenliği, E_j , $E_j = \max(0, d_j - C_j)$ ile, j işinin geçliği, T_j , $T_j = \max(0, C_j - d_j)$ ile tanımlanır. Her işe ilişkin birim erkenlik cezası α_j ve birim geçlik cezası β_j vardır. Ceza fonksiyonlarının doğrusal olduğu varsayılarak, bir S çizelgesi

için standart E/G amaç fonksiyonu $f(S) = \sum_{j=1}^n (\alpha_j E_j + \beta_j T_j)$ olarak yazılabilir. Bazı

durumlarda teslim tarihinin belirlenmesine ilişkin bir maliyet, δ_j , söz konusu olabilir. Bundan dolayı, j işi bir teslim tarihi cezası $\delta_j d_j$ alabilir ve E/G problemi

$f(S) = \sum_{j=1}^n (\alpha_j E_j + \beta_j T_j + \delta_j d_j)$ şeklindeki daha genel amaç fonksiyonuna

genişletilebilir. Baker ve Scudder [4] tarafından E/G problemleriyle ilgili kapsamlı bir inceleme yapılmıştır.

E/G probleminde, ya tüm işler için aynı (ortak) ya da işten işe değişen (farklı) teslim zamanları kararlaştırılır. Farklı teslim tarihleri olan problemlerin çözümü ortak teslim tarihli problemlerin çözümünden farklıdır. İşler bağımsız ceza ağırlıklarına sahipse $\alpha_j = \alpha$ ve $\beta_j = \beta$ alınır. Aksi takdirde problem, işe-bağımlı ceza ağırlıklı problem tipine dönüşür.

2.1. Ortak Teslim Tarihli Modeller

2.1.1. Ortak teslim tarihinden toplam mutlak sapmanın en küçüklenmesi

Ortak teslim tarihinden işlerin tamamlanma zamanlarının toplam mutlak sapmasının en küçüklenmesi E/G problemlerinin önemli bir özel durumudur. Bu durumda, tüm

d_j 'ler d 'ye eşittir. Eğer d yeterince büyükse, problem kısıtlandırılmamış aksi takdirde ise kısıtlandırılmış ortak teslim tarihli problemdir. Kısıtlandırılmamış teslim tarihli bir problem için en iyi çözüm, aşağıdaki özellikler dâhilinde ortaya çıkar [4].

Özellik I: En iyi çizelgede boş zaman yoktur.

Özellik II: En iyi çizelge V biçimindedir, yani teslim tarihinden sonra biten işler, işlem süresinin artan sırasına göre sıralanırken, teslim tarihinde veya öncesinde biten işler, işlem süresinin azalan sırasına göre sıralanır.

Özellik III: En iyi çizelgede, teslim tarihinde tamamlanan bir iş vardır.

B , teslim tarihinde ya da teslim tarihinden önce tamamlanan işler kümesini göstermek üzere erken biten işlerin sayısı $b = |B|$ ve A , teslim tarihinden sonra tamamlanan işler kümesini göstermek üzere geç biten işlerin sayısı $a = |A|$ 'dır. B_i , B 'deki i . işin indeksini ve A_i , A 'daki i . işin indeksini gösterebilir. Bazı cebirsel değişikliklerle B 'deki işlerin toplam cezası aşağıdaki gibi yazılabilir:

$$C_B = 0p_{B1} + 1p_{B2} + \dots + (b-2)p_{B(b-1)} + (b-1)p_b \quad (2.1)$$

Benzer şekilde A 'daki işlerin toplam cezası şöyledir:

$$C_A = ap_{A1} + (a-1)p_{A2} + \dots + 2p_{A(a-1)} + 1p_a \quad (2.2)$$

Amaç fonksiyon C_B ve C_A 'nın toplamıdır ve işlem süreleri bellidir. a ve b bilindiğinde, en büyük işlem süreli olan toplamdaki en küçük katsayıyla sonraki en büyük işlem süreli en küçük katsayının karşılaştırılmasıyla bu toplam en küçüklenebilir. Bu karşılaştırma bu şekilde devam eder, eşitlik durumunda keyfi olarak biri seçilir. Eğer n çift bir sayı ise $b = a$, eğer n tek bir sayı ise $b = a + 1$ olmalıdır. Böylece aşağıdaki özellik elde edilir:

Özellik IV: En iyi çizelgede, b . sırada yer alan iş d 'de tamamlanır. b , $n/2$ 'ye eşit veya daha büyük en küçük tamsayıdır.

En iyi çizelgelerin sayısı 2^r 'dir [5]. Burada,

$$r = (n - 1)/2 \quad \text{eğer } n \text{ tek ise} \quad (2.3)$$

$$r = n/2 \quad \text{eğer } n \text{ çift ise} \quad (2.4)$$

Bagchi ve arkadaşları [6] ikincil ölçüt olarak B kümesindeki işlem sürelerinin toplamını en küçüklemeyi dikkate almışlardır. İşleri derecelendiren ve onları derecelendirilmiş listeden sırasıyla pozisyona atayan sisteme Algoritma 1 adı verilir. Bu düzenlemede B 'nin uzunluğu $\Delta = p_n + p_{n-2} + p_{n-4} + \dots$ ile hesaplanır. İşler işlem sürelerinin azalan sırasına göre dizildiğinde p_n en uzun işlem süreli işi gösterir. $d \geq \Delta$ ise kısıtlandırılmamış problem, $d < \Delta$ olduğunda da kısıtlandırılmış problem ortaya çıkar. d , bir karar değişkeni olursa $d = \Delta$ olarak belirlenir ve Algoritma 1 uygulanır. Kısıtlandırılmış durumda Özellik I ve II sağlanır (Özellik II'nin ispatı Raghavachari [7] tarafından yapılmıştır). Bununla birlikte Özellik III sağlanmaz, en iyi çözümde d 'den önce başlayıp, d 'den sonra biten bir iş olabilir. Dolayısıyla Özellik IV de sağlanmaz. Hall ve arkadaşları [8], kısıtlandırılmış problemin NP-tam olduğunu göstermiştir.

Ortak teslim tarihinden toplam mutlak sapmanın en küçüklenmesi ölçütünü inceleyen ilk makale olarak Kanet'in [9] çalışması dikkate alınır. İlgilendiği problem kısıtlandırılmamış ortak teslim tarihli ($d \geq \sum p_j$) tek-makine E/G problemidir. Erkenlik ve geçlik ceza ağırlıklarının simetrik ($\alpha_j = \beta_j$) ve işlerden bağımsız ($\alpha_j = \alpha$ ve $\beta_j = \beta$) olduğu varsayılmıştır. Problemin amaç fonksiyonu aşağıdaki gibi yazılabilir:

$$f(S) = \sum_{j=1}^n (E_j + T_j) \quad (2.5)$$

İşlem sürelerinin azalan sırasında, erken işler kümesinin en sağındaki pozisyona ve geciken işler kümesinin en solundaki pozisyona işleri aşamalı olarak yerleştirerek en iyi çözümü veren bir algoritma önermiştir. Algoritmaya göre, V-biçimlilik özelliği sağlanır ve ortak teslim tarihi erken işler kümesinin en sağındaki işin tamamlanma zamanına eşittir.

Bagchi ve arkadaşları [5] kısıtlandırılmış problemi çözmek için bir algoritma taslağı oluşturmuştur. Çizelgenin başlama zamanının sıfır olduğu varsayılmıştır. Szwarc [10] en uygun başlama zamanının sıfır olmayabileceğine işaret etmiştir, bu nedenle Bagchi ve arkadaşlarının [5] algoritması en iyi çözümü bulmayı garanti etmez. Szwarc [10] bir yeterlilik koşulu geliştirerek çizelgenin hangi durumda sıfır zamanında başlaması gerektiğini belirlemiştir. Buna göre eğer $p_1 + p_2 + \dots + p_r \geq d$ koşulu sağlanıyorsa çizelgenin sıfır zamanında başlayacağı varsayılır. Bu durumda da problemi çözmek kolay değildir. Hem Bagchi ve arkadaşları [5] hem de Szwarc [10] birerleme yönetimini uygulamıştır. Sundararaghavan ve Ahmed [11] ise başlama zamanının sıfır olması durumunda etkili bir şekilde çalışan sezgisel bir çözüm yöntemi geliştirmiştir. Baker ve Chadowitz [12] bu koşulu yumuşatarak algoritmayı genelleştirmiştir.

Rabadi ve arkadaşları [13] araştırmalarında erkenlik ve geçliği en küçükleme amacı taşıyan tek-makine çizelgeleme problemini ele almıştır. Erkenlik ve geçlik eşit ağırlıklara sahiptir ve teslim tarihi her iş için ortak ve kısıtlandırılmamıştır. Sıraya-bağımlı hazırlık sürelerini içeren problem anlamlı bir şekilde büyümekte ve NP-zor olmaktadır. Literatürde problemi en iyi şekilde çözmek için sadece karışık tamsayılı programlama (KTP) modelleri mevcuttur. Rabadi ve arkadaşları [13] bu çalışmalarında, problem için en iyi çözümler elde eden bir dal-ve-sınır algoritması (D&S) geliştirmişlerdir. Bu makalenin literatüre katkısı, D&S'nin makul süreler içinde 25 işe kadarki bu tip E/G problemini çözebilmesidir. Literatürde KTP kullanılarak sadece 8 işe kadarki problemler en iyi şekilde çözülebilmekteydi. D&S için geliştirilen alt sınır, üst sınır ve dallanma stratejisi test edilmiş ve etkili olduğu gösterilmiştir. Her düğümde alt sınırı hesaplamak uzun vakit olsa da alt sınırın kalitesi D&S'nin performansını büyük ölçüde etkiler ve daha iyi alt sınırlar D&S'nin sonuçtaki performansının daha iyi olması anlamına gelir. D&S'nin sonuçtaki performansını test etmek için deney tasarımında iş sayısı, düzenlenmiş işlem süreleri aralığı ve üst sınır kullanılmıştır. Tasarım sonucunda iş sayısı en anlamlı faktör olarak çıkmıştır.

2.1.2. Farklı erkenlik ve geçlik cezaları

Farklı erkenlik ve geçlik cezaları söz konusu olduğunda α 'nın üreticinin ve β 'nin tüketicinin belirlediği katsayılar olması muhtemeldir. Bu takdirde amaç fonksiyonu

$$f(S) = \sum_{j=1}^n (\alpha E_j + \beta T_j) \text{ şeklinde olur. Bu problem, Bagchi ve arkadaşları [6] ile}$$

Emmons [14] tarafından analiz edilmiştir. Kısıtlandırılmış ve kısıtlandırılmamış olma durumu yine söz konusudur. Kısıtlandırılmamış durum için ilk üç özellik aynıdır ve dördüncü özellik aşağıdaki gibi olur:

Özellik IV: En iyi çizelgede sıradaki b . iş d' de tamamlanır. Burada b , $n\beta/(\alpha + \beta)$ 'ya eşit ya da bu değerden daha büyük en küçük tamsayıdır.

Problemin NP-tam olan kısıtlandırılmış hali için Bagchi ve arkadaşları [6] Özellik I ve II'nin sağlandığını göstermiştir. Bununla birlikte, sıfır zamanında başlama varsayımının kullanılması dışında kısıtlandırılmış durum için en iyileyici bir yöntem geliştirilmemiştir. Baker ve Chadowitz [12] bu durum için sezgisel bir yaklaşım sunmuştur.

2.1.3. İlave cezalar

Temel E/G ölçütünü genişletmek için performansın, diğer cezaların değerlendirildiği boyutları probleme ilave edilir. Böyle iki boyut, yani teslim tarihi cezası ve akış zamanı cezası, Panwalkar ve arkadaşları [15] tarafından tanımlanmıştır. Araştırmacılar ortak teslim tarihine göre ceza verilen tek makineli bir E/G problemi çalışmıştır. Problemleri, işlerin ceza ağırlıklarına bağımlı olduğunu varsayar. δ katsayısı, her zaman birimi için teslim tarihi atama maliyetini ve d_0 parametresi, kabul edilebilir en büyük teslim tarihini göstermek üzere amaç fonksiyonu

$$f(d, S) = \sum_{j=1}^n [\alpha E_j + \beta T_j + \delta(d - d_0)^+] \text{ olarak yazılabilir. Panwalkar ve arkadaşları}$$

[15] bu problemin birerleme dışında bir yöntemle çözülemeyeceğini ifade etmiştir. Tek istisna d_0 'ın sıfır olduğu özel durumdur. Bu durumda amaç fonksiyonu

$f(d, S) = \sum_{j=1}^n (\alpha E_j + \beta T_j + \delta d)$ olarak yazılabilir. Bu problem için Özellik I, II ve III

sağlanır. Özellik IV aşağıdaki gibi olur:

Özellik IV: En iyi çizelgede sıradaki b . iş d' 'de tamamlanır. Burada b , $n(\beta - \delta)/(\alpha + \beta)$ 'ya eşit ya da bu değerden daha büyük en küçük tamsayıdır.

Panwalkar ve arkadaşları [15] pozisyona göre ceza kavramını kullanarak problemi en iyi şekilde çözmek için bir algoritma sunmuştur. Erken işler kümesindeki işlerin pozisyona göre cezaları $\{n\delta + 0\alpha, n\delta + 1\alpha, n\delta + 2\alpha, \dots\}$ ve geciken işler kümesindeki işlerin pozisyona göre cezaları $\{\dots, 3\beta, 2\beta, 1\beta\}$ şeklinde yazılabilir. Algoritma daha uzun işlem süreli bir işi daha küçük cezalı bir pozisyona yerleştirir. Başka bir deyişle, en uzun işlem süreli iş, en soldaki pozisyona yerleştirilmelidir veya ilk olarak işlenmelidir.

Chang ve Joo [16] işlerin değişik zamanlarda hazır olmalarına izin veren ortak teslim tarihli bir tek-makine E/G problemini ele almıştır. Her iş isteğe bağlı bir zamanda işlem sürecine başlayabilir fakat iş belirlenen hazır olma zamanından önce işlenmeye başladığında başlama zamanı cezasına maruz kalır. Amaç, erkenlik, geçlik ve başlama-zamanı cezalarının toplamını en küçükleyen çizelgeyi bulmaktır. Araştırmacılar, belli bir sıra için her işin en iyi başlama zamanını belirlemek amacıyla bir algoritma önermişlerdir. Probleme iyi bir çözüm bulmak için de en iyi başlama zamanı algoritmasının kullanıldığı bir sezgisel arama sunulmuştur.

Panwalkar ve Rajagopalan [17] işlerin işlem sürelerinin doğrusal bir maliyetle kontrol edilebilen değişkenler olduğu statik bir tek-makine E/G problemini incelemiştir. Problemlerinde, bir işin işlem süresi normal bir değere sabitlenebilir. Bununla birlikte, ilave maliyetler ödeyerek, işlem süresini azaltmak için makine daha yüksek hızda çalıştırılabilir. j işinin işlem süresini azaltmanın birim maliyeti G_j olarak verildiğinde, x_j , sıkıştırılan j işinin gerçek zaman miktarını, $[j]$ ise herhangi bir sıra, S , için j pozisyonundaki işi gösterir. Böylece problemin amaç

fonksiyonu $f(d, S) = \sum_{j=1}^n (\alpha E_{[j]} + \beta T_{[j]} + G_{[j]} x_{[j]})$ olarak yazılabilir. Araştırmacılar, bir $n \times n$ maliyet matrisi oluşturarak problemin bir atama problemi olarak en iyi şekilde çözülebileceğini göstermiştir.

Liman ve Ramaswamy [18] bir ‘Erkenlik/Geçlik Aralık’ problemi sunmuştur. Modellerinde, tüm işlerin $d_2 - d_1$ ölçüsünde ortak bir dağıtım aralığına sahip olduğu varsayılır. İşler dağıtım aralığı dâhilinde tamamlanırsa maliyete sebep olmaz. Kesintiye izin verilmez ve tüm işlerin sıfır zamanında mevcut bulunduğu varsayılır. j işinin belirlenmiş cezası, γ_j , verildiğinde problemin amacı toplam ceza

fonksiyonunu $(f(S) = \sum_{j=1}^n [\alpha_j \max\{0, d_1 - C_j + \gamma_j U_j\}])$ en küçükleyen çizelgeyi

belirlemektir. Çalışmada teslim tarihinin hem kısıtlandırılmış (d_1 ve d_2 önceden biliniyor) hem de kısıtlandırılmamış (d_1 ve d_2 karar değişkenleridir) olduğu durumlar dikkate alınmıştır. Araştırmacılar problemin NP-tam olduğunu ispatlamış ve her iki durumu çözmek için dinamik programlama algoritmaları sunmuştur.

Her iş için geçlik cezasının belirlenmiş olduğunu varsayan modelin aksine, Liman ve arkadaşları [19] zamanın bir fonksiyonu olarak hem erkenlik hem de geçlik cezalarını içeren problemi çalışmıştır. Problemlerinde, ortak teslim aralığı, D , belirli bir parametredir. Birim zamandaki erkenlik cezası (α), birim zamandaki geçlik cezası (β) ve teslim aralığının konum cezasının (δ) sabit olduğu varsayılır. Amaç, erkenlik, geçlik ve ortak teslim aralığı konum cezalarının ağırlıklı toplamını en küçükleyen iş çizelgesini ve ortak teslim aralığı konumunu belirlemektir. Araştırmacılar, problemin özelliklerini araştırmış ve problemi en iyi şekilde çözebilmek için polinom zamanlı bir algoritma önermiştir. Basit çözümlerden elde edilebilen iki durumu da dikkate almışlardır. Birincisi $\delta \geq \beta$ olma durumudur. Bu durumda, teslim aralığının konumu ve en iyi çizelge, $d_1=0$, $d_2=D$ alınmasıyla ve işlerin, işlem sürelerinin artan sırasına göre dizilmesiyle belirlenebilir. İkincisi $(\beta + \alpha)/(\delta + \alpha) \geq 0$ olma durumudur. Teslim

aralığının konumu ve en iyi çizelge $d_2 = \sum_{j=1}^n p_j$ ve $d_1 = d_2 - D$ alınmasıyla ve işlerin,

işlem sürelerinin azalan sırasına göre dizilmesiyle belirlenebilir.

Ventura ve arkadaşları [20] kaynağa-bağımlı geliş tarihleri ve kısıtlandırılmamış ortak teslim tarihli bir tek-makine E/G problemini çalışmıştır. Bir işin kaynak tüketimi maliyetinin, işin geliş tarihinin azalan bir doğrusal fonksiyonu olduğu ve bu fonksiyonun tüm işler için ortak olduğu varsayılmıştır. Amaç, toplam kaynak tüketimi ile erkenlik ve geçlik cezalarını en küçükleyen çizelgeyi ve işlerin geliş tarihlerini bulmaktır. Problemin NP-zor olduğu gösterilmiştir. Küçük ve orta büyüklükteki problemler için bir dinamik programlama geliştirilmiştir. Büyük boyutlu problemler için de bir sezgisel algoritma önerilmiş ve sezgisel ile en iyi çözümler arasındaki işlemsel karşılaştırma değerlendirilmiştir. En iyi çözümlerin yapısını karakterize etmek ya da önerilen sezgisel algoritma için gelişmiş çözümleri bulmak için bazı özellikler gösterilmiştir. En iyi çözümü bulma garantisi olmamasına rağmen önerilen sezgisel algoritma yüksek kalitede çözümler sağlayabilmiştir.

2.1.4. Doğrusal olmayan cezalar

Bazı durumlarda teslim tarihinden büyük sapmalar kesinlikle istenmez. Bu nedenle performans ölçütü olarak ortak teslim tarihinden sapmaların karelerinin kullanılması

daha uygun olabilir ve amaç fonksiyon $f(S) = \sum_{j=1}^n (d - C_j)^2 = \sum_{j=1}^n (E_j^2 + T_j^2)$ haline

gelir. Bagchi ve arkadaşları [21] bu problemin kısıtlandırılmamış halinin, Eilon and Chowdhury [22], Kanet [23] ve Vani ve Raghavachari [24] tarafından çalışılan tamamlanma zamanı varyansı problemine denk olduğunu göstermiştir. Eilon and Chowdhury [22] Özellik II'yi ispatlamış ve karesel problemi çözmek amacıyla da bitişik iş çiftlerinin yer değiştirmesini kullanarak ilk sezgisel algoritmayı önermiştir. Kanet [23] problemin, işlerin tamamlanma sürelerinin toplamını en küçüklemeye denk olduğunu göstermiştir. Karesel mutlak sapma problemi için olan bir algoritmayı, bir sezgisel olarak uyarlamış ve Eilon ve Chowdhury'nin [22] sonuçlarını geliştirmiştir. Vani ve Raghavachari [24] tüm iş çiftlerinin yer değiştirmesinin kullanımını incelemiş ve diğer sezgisellerden daha iyi sonuçlar elde etmiştir. Bagchi ve arkadaşları [6] hem kısıtlandırılmış hem de kısıtlandırılmamış problem için uygulanan birerleme yöntemlerini hızlandırmak için birkaç üstünlük özelliği kullanmıştır. Bununla birlikte, kısıtlandırılmış durum için çizelgenin sıfır zamanında

başladığı varsayılmıştır. De ve arkadaşları [25] bu noktayı incelemiş ve çizelgenin sıfır zamanında başladığı varsayımını kullanmadan birerleme çözüm yöntemi geliştirmiştir.

Bagchi ve arkadaşları [6] erkenlik ve geçlik cezalarının farklı olduğu durumu da $(f(S) = \sum_{j=1}^n (\alpha E_j^2 + \beta T_j^2))$ incelemiştir. Yine üstünlük özellikleri geliştirmişler ve onları, problemi çözmek için uygulanan bir arama yöntemi ile birleştirmişlerdir. Bununla birlikte, yaklaşımları esas olarak birerleme şeklinde kalmıştır. Karesel cezaların olduğu durumda Özellik I ve II sağlanır fakat Özellik III sağlanmaz. Doğrusal olmayan cezalar durumu için Özellik IV'ün herhangi bir genelleştirilmesi mevcut değildir.

Leung [26] her birinin bir işlem süresine ve ortak teslim tarihine sahip olduğu n işin tek makine üzerinde sıralanması ile ilgili problemi dikkate almıştır. Ortak teslim tarihinin tüm işlerin teslim tarihine dek tamamlanabilecek kadar büyük olduğu varsayılmıştır. Erkenlik ve geçliğin en küçük olduğu çizelgeyi bulmak için bir $O(n \log n)$ -zaman algoritması olduğu bilinmektedir. Bu çalışmada ikili ölçütlü bir çizelgenin bulunması amaçlanmıştır. Birincil hedef, erkenlik ve geçlik toplamının en küçüklenmesidir. İkincil hedefler ise (1) en büyük erkenlik ve geçliğin; (2) en büyük erkenliğin karesi ve en büyük geçliğin karesinin toplamının; (3) erkenlik ve geçliğin karelerinin toplamının en küçüklenmesidir. İlk iki ölçüt için problemlerin NP-zor olduğu gösterilmiş ve her ikisi için tamamıyla polinom zamanlı yaklaşım planı verilmiştir. Son iki ölçüt için en kötü çizelgenin en iyi çizelgeye oranının $3/2$ 'den fazla olmadığı gösterilmiştir. Araştırmacının dikkate aldığı hedefler şöyle gösterilebilir:

Çizelge 2.1. Birincil hedef ve ikincil hedefler

Birincil Hedef	İkincil Hedefler
$t(S) = \sum_{i=1}^n (E_i + T_i)$	$f(S) = \sum_{i=1}^n (E_i^2 + T_i^2)$ $g(S) = \max_{i=1}^n \{E_i^2\} + \max_{i=1}^n \{T_i^2\}$ $h(S) = \max_{i=1}^n \{E_i, T_i\}$

2.1.5. İşe-bağımlı erkenlik ve geçlik cezaları

E/G cezaları işe-bağımlı olarak değiştiğinde amaç fonksiyon

$f(S) = \sum_{j=1}^n (\alpha_j E_j + \beta_j T_j)$ olarak yazılabilir. Bu problemin kısıtlandırılmamış durumu

Cheng [27], Quaddus [28], Bector ve arkadaşları [29] ve Hall ve Posner [30] tarafından incelenmiştir. Cheng [27] temel problem için Özellik IV'ün varyasyonunu çıkarmıştır. Bector ve arkadaşları [29] aynı sonuçlar üzerinden doğrusal programlama perspektifi sunmuştur. Hall ve Posner [30] Özellik I ve II'nin ilgili varyasyonlarını ve en iyi sıra için gerekli koşulları sağlayan bazı üstünlük özelliklerini ispatlamıştır. En önemli sonuçları ise problemin kısıtlandırılmış halinin NP-tam olduğunun ispatıdır. Sahte polinom olduğunu gösterdikleri dinamik programlama algoritmasını geliştirmeye devam etmişlerdir. Ek olarak, yüzlerce iş içeren problemleri ele alarak ve makul koşturma sürelerinde en iyi sonuçları elde ederek algoritmalarının işlemsel etkililiğini göstermişlerdir.

Quaddus [28] $\alpha_j \neq \beta_j$ olan ve teslim tarihi cezası, δ_j , içeren genel durumu incelemiş fakat sadece teslim tarihi seçimi ile ilgilenmiştir. Baker ve Scudder [4] problemin sıralama yönünü ihmal ettiğini belirtmiştir. Bununla birlikte, Özellik I'in hala geçerli olduğunu göstermek oldukça kolaydır ve Özellik II şu şekle dönüşmüştür:

Özellik II: En iyi çizelge V-biçimlidir, B 'deki işler p_j/α_j oranının azalan sırasına göre sıralanır; A 'daki işler p_j/β_j oranının artan sırasına göre sıralanır.

Buna ilaveten Özellik III sağlanır ve Özellik IV şu şekilde olur:

Özellik IV: En iyi çizelgede sıradaki b . iş d 'de tamamlanır. Burada b ,

$$\sum_{j=1}^b (\alpha_j + \beta_j) \geq \sum_{j=1}^n (\beta_j - \delta_j) \text{ eşitsizliğini sağlayan en küçük tamsayıdır.}$$

Hino ve arkadaşları [31] ortak teslim tarihli bir tek-makine problemi incelemiştir. Performans, işlerin erkenlik ve geçlik cezalarının ağırlıklı toplamının en küçüklmesi ile ölçülmüştür. Bu problem NP-zor olduğu için tabu arama (TA) tabanlı bir sezgisel ve bir genetik algoritma önerilmiştir. Bu metotların performansını artırmak için melez stratejiler de analiz edilmiştir. Önerilen yaklaşımlar Biskup ve Feldmann'dan [32] seçilen 280 standart problem ile karşılaştırılmıştır. Problemler setinde yedi farklı sayıda iş grubu ve ortak teslim tarihini belirlemek için dört kısıtlayıcı faktör, h , bulunmaktadır. Ortak teslim tarihi, kısıtlayıcı faktör ve işlem sürelerinin toplamının çarpımıyla elde edilir ($d = \left\lfloor h \sum_{j=1}^n p_j \right\rfloor$). Elde edilen sonuçların kalitesiyle ilgili olarak, kurucu sezgiselin standart çözümlerin en az %43'üne ulaştığı söylenebilir. Meta sezgisellerin kullanımı daha yüksek kaliteli sonuçlara ulaştırmıştır. Her iterasyonda boş zaman faktörünün düzenlemesi, meta sezgisellerin daha iyi sonuçlara ulaşmasında en büyük etkidir. TA algoritması standart değerlerin en az %77'sine ulaşmıştır. Genetik algoritma ve melez stratejiler ile elde edilen sonuçların kalitesi benzerdir, ancak melez stratejilerin işlemsel süresi anlamlı bir şekilde daha düşüktür. Önerilen algoritma ile çözülen 140 problemin 120'sinde Feldmann ve Biskup [33] tarafından bulunan sonuçlara eşit ya da onlardan daha iyi sonuçlara ulaşılmıştır.

2.1.6. Paralel makineli modeller

Kısıtlandırılmamış durumun temel analizi paralel makineleri içeren modellere genişletilmiştir. Hall [34] ile Sundararaghavan ve Ahmed [11] paralel çalışan m

özdeş makineli sistemde toplam mutlak sapmanın en küçüklenmesi konusunda çalışmıştır. Sundararaghavan ve Ahmed [11], Kanet'in [9] çalışmasını m -paralel özdeş makineye genişletmiştir. Problemlerinde kısıtlandırılmamış ortak teslim tarihi ele alınmıştır. En iyi çözümü veren iki-adımlı bir algoritma geliştirmişlerdir. İlk adım, makinelerin m alt kümelerindeki işlerin kümesi için en iyi bölmeyi bulmaktır. İkinci adım, Kanet'in [9] algoritmasını kullanarak her bir makinedeki işlerin çizelgesini belirlemektir. İlk işin sıfır zamanında başlamak zorunda olduğu varsayımına dayalı olarak, araştırmacılar kısıtlandırılmış ortak teslim tarihli tek-makine problemini çözmek için bir de sezgisel sunmuşlardır.

Emmons [14] farklı erkenlik ve geçlik cezaları için bu analizi genişletmiştir. Ek olarak, özdeş olmayan paralel işlemcileri ele almıştır. Çoklu makine yöntemi, en kısıdan en uzuna katsayıların ve işlem sürelerinin eşlenmesini uygulayarak bu duruma kolaylıkla adapte edilmiştir. Emmons [14] ayrıca, en iyi sonuçlar için son işin tamamlanma zamanının en küçüklenmesi veya en küçük teslim tarihinin elde edilmesi gibi ikincil ölçütleri ele almış ve özel durumlar için çözümler sağlamıştır.

Panwalkar ve arkadaşları [15] tarafından incelenen problem, Cheng [35] tarafından paralel makine durumuna genişletilmiştir. Amacı, erkenlik, geçlik ve teslim tarihi cezalarını en küçükleyen m paralel makinedeki çizelgeyi belirlemektir. Paralel makineli sistemler için en iyi teslim tarihinin sıraya bağımlı olduğunu göstermiş ve problemi çözmek için bir sezgisel incelemiştir. Bu sezgisel, paralel makineleri sanki m bağımsız tek makineymiş gibi dikkate almıştır. İşler, Panwalkar ve arkadaşlarının [15] çalışmasına benzer şekilde pozisyona göre ceza kavramını kullanarak atanmıştır.

Adamopoulos ve Pappis [36] ortak teslim tarihi altında m paralel bağlantısız makinedeki n bağımsız işin çizelgelendiği problemi çalışmıştır. Modelde, teslim tarihi karar değişkenidir. Hem erkenlik hem de geçlik cezaları işlerden bağımsızdır. Her iş makineden makineye değişen bir işlem süresi gerektirir. i makinesindeki j işinin erkenliği ve geçliği $E_{[i,j]}$ ve $T_{[i,j]}$ şeklinde alınırsa, ele aldıkları problem

$$f(d, S) = \sum_{i=1}^m \sum_{j=1}^n (\alpha d + \alpha E_{[i,j]} + \beta T_{[i,j]})$$

amaç fonksiyonunu en küçükleyen en iyi ortak

teslim tarihini ve m makine üzerindeki n işin çizelgesini belirlemektir. Algoritma dört farklı safhaya bölünebilir. Birinci safha m makineye n işin tahsis edilmesi problemini ele alır. Çatışan iki ölçüt dikkate alınır. İlki, her işin, işlem süresinin mümkün olduğunca küçük olduğu bir makineye atanmasıdır. İkincisi, farklı makinelerdeki en büyük akış zamanları arasındaki geniş farklılıklardan kaçınmaktır. İkinci safha bir makine üzerindeki n kümesinin sırasını belirlemek amacıyla araştırmacı tarafından sunulan pozisyon etiketi kavramını kullanarak işleri pozisyonlarına tahsis etmektir. Üçüncü safhada, her makinedeki işlemin başlama zamanıyla birlikte bir ortak teslim tarihi belirlenir. Son olarak, çizelgeden elde edilen amaç fonksiyonu hesaplanır.

Biskup ve Cheng [37] erkenlik, geçlik ve tamamlanma zamanı cezalarından oluşan amaç fonksiyonunu en küçükleme amacıyla m paralel özdeş makine üzerinde n işin çizelgelenmesi problemini çalışmıştır. Problemden karar değişkeni olarak kısıtlandırılmamış ortak teslim tarihi alınmış ve araştırmacılar problemin NP-zor olduğunu kanıtlamıştır. Problemi çözmek için bir sezgisel sunulmuştur. Sezgisel, makineye düzgün bir şekilde işlerin atanmasıyla başlar. Düzgün bir şekilde işlerin atanmasıyla mümkün olduğunca çok iş yükü eş zamanlı olarak dengelenmeye çalışılır. m tane en büyük iş makinelerine her atandığında, tüm makinelerin iş yükü hesaplanır ve en uzun iş, en küçük iş yüklü makineye atanır. İşleri makinelerine atadıktan sonra, m -tek makine problem bağımsız olarak çözülür. m -tek makine probleminin çözümüyle farklı makineler için farklı teslim tarihleri elde etmek olasıdır. Bu nedenle, sezgiselin son adımında, her makinenin ilk işinin başlama zamanı ve ortak teslim tarihi, bir doğrusal programlama probleminin çözülmesiyle eş zamanlı olarak hesaplanır.

Panwalkar ve Liman [38] makine çalıştırma maliyeti olan, tek operasyonlu E/G çizelgeleme problemini ele almıştır. Problemlerinde, sıfır zamanında işlenmeyi bekleyen n iş vardır. İşin yapılması için çalıştırılmaya hazır çok sayıda özdeş makine vardır. Çalıştırılan her makine belirli bir makine çalıştırma maliyeti, χ , meydana getirir. Bir makinede çizelgelenmiş işler için, makineden makineye değişen bir ortak teslim tarihi vardır. k , ($1 \leq k \leq n$) kullanılan özdeş makinelerin sayısını, B_i , i makinesinde işlenecek bir yığına giren işlerin kümesini gösterir ve d teslim

tarihlerinin $(d_1, d_2, \dots, d_k)$ kümesidir. Amaç çizelgeyi, S , bulmaktır. Kullanılan makinelerin sayısı k ve her makinenin ortak teslim tarihi d olmak üzere

$$f(S, k, d) = k\chi + \sum_{i=1}^k \sum_{j \in B_i} (\alpha E_j + \beta T_j) \text{ fonksiyonunu en küçükleyen } d_i, B_i \text{ yığınının}$$

teslim tarihidir. Araştırmacılar tarafından sunulan problemin çözümü için k makinesine pozisyona göre ceza kavramı uygulanmıştır. Her makine aynı pozisyona göre ceza modeline sahiptir. Buna göre teslim tarihinden önce $\{0, 1\alpha, 2\alpha, 3\alpha, \dots\}$ ve teslim tarihinden sonra $\{\dots, 3\beta, 2\beta, 1\beta\}$ modelleri kullanılır. Makalede önerilen algoritmada $k = 1$ olarak belirlenir, pozisyona göre ceza hesaplanır, işlerin pozisyona göre aldıkları cezaları sıralanır ve toplam maliyet hesaplanır. Böylece, k 'nın değeri her iterasyonda bir artacak ve hesaplama tekrar yapılacaktır. Algoritma k ve $k-1$ makinelerinin toplam maliyetlerinin farkı en fazla δ 'ya eşit oluncaya kadar tekrarlanır. En uygun makine sayısı $k-1$ 'dir.

2.1.7. Teslim tarihi toleransları

Genel yaklaşıma göre, tamamlanma zamanı teslim tarihine yeterince yakınsa cezanın sıfır olmasına izin verilir. Burada yeterince yakınlıkla kastedilen belli bir tolerans miktarıdır. j işinin ceza almaması için tamamlanma zamanının $d-u_j$ ve $d+v_j$ aralığına düşmesi gerekir. Cheng [39] ölçütün, toplam mutlak sapma olduğu ve bütün u_j ile v_j 'lerin özdeş olduğu özel durumu analiz etmiş ve kısıtlandırılmamış durumda Özellik I ve II'nin doğrudan, Özellik III ve IV'ün de küçük değişikliklerle sağlandığını göstermiştir. Fakat değişik bir varsayım kabul etmiştir. Model, teslim tarihinin her iki tarafındaki tolerans aralığında tamamlanan bir iş için ceza tanımlamamasına rağmen diğer işler için erkenlik ve geçlik, tolerans aralığının sonundan değil teslim tarihinden hesaplanır. j işi için daha geleneksel ve uygun bir varsayım dikkate alınırsa erkenlik ve geçlik sadece tolerans aralığının sonundan ölçülür. Böylece

$$E_j = (d - C_j - u_j)^+ \quad (2.6)$$

$$T_j = (C_j - d - v_j)^+ \quad (2.7)$$

ve

$$f(S) = \sum_{j=1}^n (\alpha_j E_j + \beta_j T_j) \quad (2.8)$$

olur.

Toleranslı modelde Özellik I ve II sağlanmaya devam eder. Özellik III'ün genelleşmiş hali, en iyi çözümde herhangi bir cezaya uğramayan bir b işi olduğunu ifade eder. Özellik IV'ün genelleşmiş hali en iyi sırada b için gerekli bir koşul sağlar [4].

Özellik III(G): En iyi çizelgede, j işi $d-u_j$ ya da $d+v_j$ 'de tamamlanır.

Özellik IV(G): En iyi çizelgede b , geçlik cezasına uğramayan işlerin sayısını gösterebilir. b işinin tamamlanma zamanı aşağıdaki koşulları sağlar.

$$C_b = d - u_b \quad \text{eğer} \quad \sum_{i < b} \alpha_i < \sum_{i \geq b} \beta_i \quad \text{ve} \quad \sum_{i < b} \alpha_i \geq \sum_{i > b} \beta_i \quad (2.9)$$

$$C_b = d + v_b \quad \text{eğer} \quad \sum_{i < b} \alpha_i < \sum_{i > b} \beta_i \quad \text{ve} \quad \sum_{i \leq b} \alpha_i \geq \sum_{i > b} \beta_i \quad (2.10)$$

2.2. Farklı Teslim Tarihli Modeller

Genel E/G modelinde, her iş kendine ait bir teslim tarihine sahip olabilir. Bu özellik ortak teslim tarihli probleme göre en küçük maliyetli çizelgeyi belirlemek açısından daha fazla zorluk taşımaktadır.

Seidmann ve arkadaşları [40] farklı teslim tarihli tek-makine E/G problemini ele almıştır. Modellerinde, teslim tarihleri karar değişkenleri olarak davranır. Amaç, tedarik zamanı, erkenlik ve geçlik cezalarının toplamını en küçükleme. Tüm işler, işlem sürelerinin azalan sırasına göre dizilir. A_i , tedarik zamanını göstermek üzere $A_i = \max(0, d_i - A)$ şeklinde hesaplanır. Problemin amaç fonksiyonu

$$f(S, d) = \sum_{j=1}^n (\chi A_j + \alpha E_j + \beta T_j) \quad \text{olarak yazılabilir. Burada } \chi, \alpha \text{ ve } \beta \text{ sırasıyla}$$

birim tedarik zamanı, birim elde tutma maliyeti ve geçlik cezasını gösterir.

Araştırmacılar, en kısa işlem süreli işin ilk işlenmesine dayalı dağıtım kuralını uygulayarak en iyi çözüm veren bir algoritma sunmuşlardır. $\chi \leq \beta$ ise her işin teslim

tarihi $d_j^* = \sum_{i=1}^j p_i$ olarak belirlenir. Aksi takdirde, her işin teslim tarihi

$d_j^* = \min\{A, \sum_{i=1}^j p_i\}$ olarak belirlenir.

Garey ve arkadaşları [41] bu problemin NP-tam olduğunu ilk olarak gösteren araştırmacılarıdır. Çözüm yöntemleri içeren diğer çalışmalardan bazıları Yano ve Kim [42], Abdul-Razaq ve Potts [43] ve Ow ve Morton'a [44, 45] aittir. Bu modelde Özellik I ve II sağlanmaz. En iyi sıra V-biçimli olmayabilir ve en iyi çizelge için arama iki alt probleme ayrıştırılabilir: İyi bir iş sırası bulma ve boş zamanı yerleştirme.

2.2.1. Boş zamanların yer almadığı modeller

Abdul-Razaq ve Potts [43] teslim tarihi cezası içeren bu tip problemi çözmüş fakat sadece boş zamanların yer almadığı çizelgeleri dikkate almıştır. D&S önermişler ve iyi sınırlar elde etmek için yumuşatılmış bir dinamik programlama yöntemi kullanmışlardır. İşlemsel sonuçları, 20'den fazla iş içeren problemlerin çok fazla çözüm süresi gerektireceğini göstermişlerdir.

Ow ve Morton [45], Fry ve arkadaşlarınıninkine [46] benzer bir problem çalışmıştır fakat onlarınkinde boş zaman yerleştirmeme varsayımı vardır. Problemi çözmek için sınırlı sayıdaki çözüm yollarının paralel olarak incelendiği filtreli bir direkt arama metodu sunmuşlardır. Araştırma tekniğinin amacı geri izleme olmaksızın hızlı bir şekilde aramaktır. Hesaplama çalışması aramanın nispeten küçük bir arama ağacıyla çok iyi çözümler üretebildiğini gösterir. Bununla birlikte, bu metotla en iyi çözüm garanti edilmemektedir.

Gupta ve Sen [47] çizelgede boş zaman olmayacağı varsayımı altında farklı teslim tarihli ve karesel ceza fonksiyonlu modeli incelemiştir. Açıktır ki, bu problem

özölmesi aısından oldukça zordur. Gupta ve Sen [47] problemi özmek için bir D&S tanımlamış fakat işlemsel testleri sınır hesaplamalarının çok güçlü olmadığını göstermiştir.

Valente ve Alves [48] alışmalarında tek makinede ağırlıklı erkenlik ve gelik toplamını en küçükleme problemini ele almıştır. Boş zaman içermeyen problem için filtreli ve iyileştirilmiş direkt arama algoritmaları sunmuşlar ve var olan komşuluk arama ve dağıtım kuralı sezgiselleriyle karşılaştırmışlardır. İşlemsel sonuçlar, iyileştirilmiş direkt arama algoritmalarının filtreli benzerlerinden daha üstün olduğunu ve öncelik-tabanlı filtreleme yönteminin kurallar-tabanlı alternatife göre üstünlüğünü kanıtlamıştır. En iyi özümlemler komşuluk arama algoritması ile elde edilmiştir fakat bu yöntem işlemsel olarak yoğundur ve sadece küçük veya orta büyüklükteki örneklere uygulanabilmektedir. İyileştirilmiş direkt arama sezgiseli kaliteli ve daha hızlı bir şekilde özümlemler üretmiştir, bu nedenle büyük problemleri bile özmek için kullanılabilir.

2.2.2. Boş zamanların yer aldığı modeller

Bir iş sırası bulunduğundan sonra, doğrusal bir programlama probleminin özölmesiyle boş zamanın yeri ayarlanarak en iyi izelge üretilebilir. Bununla birlikte, Garey ve arkadaşları [41] $O(n \log n)$ zamanda alışan daha basit bir yöntem oluşturmuştur. Benzer tasarımlar Fry ve arkadaşları [46] ve Yano ve Kim [42] tarafından da geliştirilmiştir.

Fry ve arkadaşları [46] farklı teslim tarihleri ve ceza ağırlıklarına sahip işleri içeren tek-makine E/G problemini alışmıştır. Araştırmacılar, amacı ağırlıklı mutlak sapmayı en küçükleme olan doğrusal bir programlamayı özerek belli bir sıraya boş zamanı yerleştiren bir algoritma geliştirmişlerdir. Ayrıca, iş sıralamasını yapan ikili yer değıştirme tekniğini ve iş pozisyonlarını belirleyen bir boş zaman yerleştirme metodu kullanarak sezgisel bir algoritma sunmuşlardır.

Yano ve Kim [49] ağırlıklar üzerinde sınırlandırması olan bir tek-makine E/G problemi incelemiştir. Burada ağırlıkların, işlerin işlem süresine oranlı olduğu varsayılır. En iyi D&S'de dikkate alınandan pek çok mümkün çözümü eleyen bazı üstünlük özelliklerini kullanan iyi bir yöntem geliştirmişlerdir. Bu yöntem, önce bir dinamik programlama kullanarak belli bir sıra için işlerin en iyi zamanlamasını belirler, sonra basit bir sınıflandırma yapar. Buna ilaveten, bazı sınıflandırma rutinlerini ve basit bir ikili yer değiştirme tekniğini birleştiren karma bir sezgisel algoritma sunmuşlardır.

Szwarc ve Mukhopadhyay [50] farklı teslim tarihli bir tek-makine E/G çizelgeleme probleminde belli bir sıraya boş zaman yerleştirmek için En Uygun Zamanlama Algoritması sunmuştur. İşleri kümeler halinde gruplayan algoritma şu kurala dayalıdır: Eğer $d_{j+1} - d_j \leq p_{j+1}$ ise j ve $j+1$ işleri aynı kümede olmalıdır. En uygun zamanlama çizelgesinde her kümenin işlerinin kesintisiz (sadece kümeler arasında boş zaman bulunur) işlenmesi gerektiğine işaret etmişlerdir. Başlangıç çizelgesinde ilk iş sıfır zamanında başlar ve tüm işler sürekli bir şekilde işlenir.

2.2.3. Farklı özelliklerdeki diğer problem tipleri

Keyser ve Sarper [51] hazır olma zamanlarının sıfır olmadığı ve işlem öncesi bekleme süresi için cezaların olduğu tek-makine E/G problemini dikkate almıştır. w_i , j işinin birim bekleme süresi maliyetini göstermek üzere problemin amaç fonksiyonu

şöyle yazılabilir: $f(S) = \sum_{j=1}^n (w_j W_j + \alpha_j E_j + \beta_j T_j)$. Araştırmacılar, (1) ilk gelen ilk

işlem görür, (2) öncelik fonksiyonu ve (3) en küçük gevşeklik olan üç ayrı sıralama kuralını uygulamış ve bitişik ikilileri yer değiştirmeyi (API) kullanan bir algoritma geliştirmiştir. Her sıralama kuralı ile birlikte her işin başlama zamanını belirlemek için iki metot uygulamışlardır. İlk metot işleri mümkün olduğunca kısa sürede çizelgelerken, ikinci metot işleri hedef başlama zamanı ile çizelgeler. Her iterasyonda her işin başlama zamanı bu iki metottan hesaplanır. Algoritmanın sonunda üç sıralama kuralıyla bulunan maliyetler birbirleriyle karşılaştırılır. En düşük maliyetli çizelge seçilir.

Soroush ve Friendall [52] farklı teslim tarihli ve stokastik işlem süreli tek-makine E/G problemini incelemiştir. Herhangi bir S çizelgesi için, C_S , toplam beklenen E/G maliyeti, $\hat{C}_S$, işlerin beklenen tamamlanma zamanından kaynaklanan toplam deterministik E/G maliyeti ve $\bar{C}_S$, işlerin standart varyasyonundan kaynaklanan toplam deterministik E/G maliyetidir. Araştırmacılar, toplam beklenen E/G maliyeti, C_S , için alt ve üst sınırların $\hat{C}_S \leq C_S \leq \hat{C}_S + \bar{C}_S$ şeklinde gösterilebileceğine işaret etmişlerdir. Alt sınır ve üst sınır maliyetlerinin hesaplanmasına ilişkin üç sezgisel önerilmiştir. Araştırmacılar bu alt sınır ve üst sınır özelliğine dayalı üç farklı sezgisel önermişlerdir. İlk sezgisel $\hat{C}_S$ 'nin alt sınır maliyetini en küçükler, ikinci sezgisel $\hat{C}_S + \bar{C}_S$ 'nin üst sınır maliyetini en küçükler ve son sezgisel $\bar{C}_S$ 'nin maliyetini en küçükler. Bunların arasından en düşük maliyetli çözüm seçilir.

Koulamas [53] keyfi zaman aralıklı bir tek-makine E/G problemini incelemiştir. Bu problemde tek bir makinede işlenmeyi bekleyen n tane iş vardır. Her iş deterministik bir işlem süresine (p_j) ve aralık uzunluğu $w_j = d_j - b_j$ olan bir zaman aralığına (b_j, d_j) sahiptir. Her iş için zaman aralığının, işin işlem süresinden daha uzun olduğu varsayılır. Eğer bir iş kendi zaman aralığında tamamıyla işlenirse (kendi zaman aralığında işin işlenmesi başlar ve biterse) hiç ceza almaz. Diğer taraftan, eğer j işinin başlama zamanı b_j 'den önce başlarsa veya j işinin tamamlanma zamanı d_j 'yi geçerse ilgili sapmayla oranlı bir erkenlik ya da geçlik cezası alır. Amaç, toplam E/G cezasını en küçükleyen çizelgeyi belirlemektir. Problemin NP-zor olduğu ispatlanmıştır. Bu problem, iyi bir iş sırası olan ve oluşturulan sıraya boş zamanı yerleştiren iki alt probleme ayrıştırılır. Alt problemin sıralamasını çözmek için araştırmacı, diğer tek makine çizelgeleme problemleri için geliştirilen sezgisellere dayalı birtakım sezgiseller sunmuştur. Belli bir iş sırasına boş zamanı en iyi şekilde yerleştirmek için bir de algoritma önermiştir.

2.3. Zorunlu Teslim Tarihli Modeller

Bu modelde teslim tarihlerinden ziyade zorunlu teslim tarihleri ele alınmaktadır. Teslim tarihleri geçlik maliyetini ihlal edebilirken zorunlu teslim tarihleri

karşılanmak zorundadır ve ihlal edilemez. Örneğin; eğer son işin tamamlanma zamanı izin verilen en son zorunlu teslim tarihini aşarsa problemin çözümü mümkün olmaz. Bununla birlikte, sonsuz β_j 'li E/G modelleri gibi modeller ve böylece problemin çeşitli özel durumları da görülebilir.

Ahmadi ve Bagchi [59] zorunlu teslim tarihlerini karşılama kısıtı altında toplam erkenliği en küçükleme amacını dikkate almıştır. Bu problemin, hazır olma zamanları ve ortak teslim tarihi kısıtı altında tüm işlerin toplam bekleme zamanını en küçüklemeye denk olduğunu göstermişlerdir. Problem NP-tam olduğu için bir D&S geliştirmişlerdir. Çözüm uzayını sınırlandırmak için en son tamamlanma zamanı algoritmasına dayalı bir alt sınır prosedürü ve en kısa kalan işlem sürelerine dayalı bir üst sınır prosedürü kullanılmıştır. Dört tane de üstünlük özelliği geliştirilmiştir. İşlemsel testlerde D&S 40 işe kadar iyi sonuç vermiştir fakat koşturma süresi, işlem sürelerinin ve zorunlu teslim tarihlerinin ilgili büyüklüklerinden etkilenmiştir.

Ağırlıklı erkenlik toplamını en küçükleme ise daha genel bir amaçtır. Ahmadi ve Bagchi [60] hazırladıkları yöntemi takip etmiş ve eşit olmayan ağırlıklar için D&S geliştirmiştir. Chand ve Schneeberger [61] benzer bir analizi işlemiş fakat problemi çözmek için dinamik bir amaç programlama geliştirmiştir. En iyiyi bulan yöntemlerin performansını, ağırlıklı erkenlik probleminin birkaç kolay durumu için geliştirdikleri bir sezgiselle karşılaştırmışlardır. Problemin NP-tam olduğu gösterilebildiği için en iyiyi bulan yöntem büyük problemler için tavsiye edilmemiş ve araştırmacılar, sezgisellerinde ileride araştırılması gereken bazı zayıflıkları belirlemişlerdir.

Bagchi [57] hem teslim tarihleri hem de zorunlu teslim tarihleri olan E/G problemini araştırmıştır. Bu modelin en büyük özelliği müşteri siparişleri kavramıdır. İşler kümesi, her biri kendi teslim tarihine ve ilgili cezalara sahip birkaç müşterinin alt gruplarına ayrılır. Varsayıma göre bir müşteri siparişi, siparişteki tüm birimler tamamlanana kadar sevk edilemez. Belli bir çizelge, σ , için amaç fonksiyon şu şekilde yazılabilir:

$$\text{Min } f(D, \sigma) = \sum_j n_j \theta_j C_j + \sum_j \sum_i (\alpha_j E_{ij} + \beta_j T_{ij}) \quad (2.11)$$

Burada

n_j : j . müşteri siparişindeki işlerin sayısını,

θ_j : j . müşteri siparişindeki her iş için birim zaman başına düşen tedarik zamanı cezasını,

C_j : j . müşteri siparişindeki son işin tamamlanma zamanını,

T_{ij}, E_{ij} : j . müşteri siparişindeki işin geçliğini ve erkenliğini,

α_j, β_j : j . müşteri siparişindeki birim erkenlik ve geçlik cezasını,

D : Müşteri siparişleri için teslim tarihlerinin vektörünü gösterir.

Bagchi [57] bu problem için basit bir algoritmanın geliştirilmesine izin verecek birkaç öneri sunmuştur. Problem ikiye ayrıştırılabilir: Tüketici siparişlerinin en iyi sırası ve her bir siparişteki işlerin en iyi sırası. Tüketici siparişlerinin en iyi sırası $P_j / n_j \theta_j$ oranının artan sırasına göre müşteri siparişlerinin sıralanmasıyla bulunur. Burada P_j , j . müşteri siparişindeki bütün işlerin toplam süresini göstermektedir. Siparişler içerisindeki işlerin sıralaması Panwalkar ve arkadaşları [15] tarafından çözülen probleme indirgenir. Bu yöntem, her bir müşteri siparişine göre en uygun teslim tarihine karar vermekle beraber işlerin en iyi sırasına da karar verir. Bagchi'nin [57] çalışmasının önemi, ortak teslim tarihli bir modelin nasıl bazı büyük problemlerin bir parçası olarak ortaya çıkabildiğini göstermesidir. Sonuç olarak, Bagchi [57] daha önceden atanmış teslim tarihlerinin bulunduğu durumlarla ilgilenmemiş fakat gelecek araştırmalar için bu problemlerin bazı ilginç uzantılarını sunmuştur.

2.4. Sıraya-Bağımlı Hazırlık Süreleri İçeren Modeller

E/G literatürünün çoğunda makinenin, işleri işlemek üzere daima hazır olduğu ve hazırlığa ihtiyaç duymadığı varsayılır. Bazı durumlarda ise hazırlık süresi işlem süresinin içinde yer alır. İkinci durum ilkinden çok farklı değildir çünkü her iki durumda da makinenin, işlerin sırasına bakılmaksızın yani sıradan-bağımsız aynı hazırlık süresini gerektirdiği varsayılmıştır. Bununla birlikte, pek çok gerçek durumda hazırlık süreleri, işlenecek olan iş kadar son işlenen işin tipine de bağlıdır yani sıraya-bağımlıdır [58]. Genel olarak, sıraya-bağımlı hazırlık süreleri içeren

izelgeleme problemleri NP-zor olan Gezgin Satıcı Problemi'ne (GSP) benzerdir [59]. Allahverdi ve arkadaşları [60] hazırlık süreleri ieren izelgeleme literatürünü incelemiştir. Makalelerinde literatürü iki kategoriye ayırmışlardır: Yığın ve yığın-olmayan problemler. Kategorilerin her birinde araştırmacılar hazırlık süreleri literatürünü sıraya-bağımlı ve sıradan-bağımsız olmak üzere iki bölümde sınıflandırmıştır. İncelemelerinde çok az sıraya-bağımlı E/G problemine değinilmiştir.

Coleman [61] tek makinede n işin izelgelenmesi iin 0/1 KTP modeli sunmuştur. Ama, yığın-olmayan sıraya-bağımlı hazırlık süreleri söz konusu olduėunda ağırlıklı erkenlik ve geliėi en küçüklemeektir. İşe-bağımlı cezalar varsayılmıştır ve teslim tarihleri farklı olarak verilmiştir. Coleman [61] tamsayılı programı test etmiş ve 8 işe kadarki problem örnekleri iin en iyi özümle bulmuştur. İşlerin sayısı arttıėında tamsayılı program yüksek işlem zamanı gerektirir. Coleman'ın [61] alışması küçük sayıdaki işler iin sıraya-bağımlı hazırlık sürelerini ieren E/G problemiyle ilgili birkaç makaleden biridir.

Chen [62] yığın sıraya-bağımlı hazırlık süreleri ieren E/G problemini ele almış ve iki problem incelemiştir. İlkinde, her bir iş yığını belli bir ortak teslim tarihi ve eşit olmayan cezalara sahiptir. Sadece iki iş yığını olması ve iki teslim tarihinin kısıtlandırılmamış derecede büyük olması durumunda problemin NP-zor olduėunu göstermiştir. Makalenin ikinci bölümünde teslim tarihlerinin karar değışkenleri olarak alınması dışında aynı problem ele alınmıştır. Araştırmacı, problem iin bazı en iyileyici özellikler sunmuş ve iki yığın işli problemi özmek iin bir polinom dinamik programlama algoritması önermiştir. Ayrıca farklı yığınlar iin ortak teslim tarihleri eşit olduėunda her iki problemin özel bir durumunu dikkate almıştır. Özel durum iin her iki problemi özmek amacıyla bir dinamik programlama algoritması oluşturmuştur.

alışmanın odaėı ortak teslim tarihli tek-makine E/G problemiyle sınırlı olacaėı iin izelge 2.2 bu problem üzerine yapılmış alışmaları özetlemiştir. Bu izelgedeki ölçütler E/G cezaları, ortak teslim tarihi ve sıraya-bağımlı hazırlık süreleridir.

Çizelge 2.2. Tek-makineli E/G problemi için literatür özeti

Referans	$\alpha_j = \beta_j = 1$	$\alpha_j = \alpha$ $\beta_j = \beta$	α_j, β_j	Kısıtlandırılmış d	Kısıtlandırılmamış d	Hazırlık Süresi
Kanet [9]	■				■	
Sundararaghavan ve Ahmed [11]	■			■		
Sundararaghavan ve Ahmed [11]	■				■	
Bagchi ve arkadaşları [5]	■			■		
Hall [34]	■				■	
Emmons [14]		■			■	
Szwarc [10]	■			■		
Hall ve Posner [63]			■	■		
Hall ve arkadaşları [64]	■			■		
Coleman [61]			■			■
Liman ve Lee [65]	■			■		
James [66]			■	■		
Chen [62]		■			■	
Chen ve Powell [67]			■		■	
Liaw [68]			■			
Rabadi [3]	■				■	■

3. HAZIRLIK SÜRELERİ İÇEREN E/G PROBLEMLERİNİN ANALİZİ

Pek çok gerçek durumda bir işi işlemekten önce makinenin hazırlanması gerekir. Sıraya-bağımlı hazırlık sürelerine yaygın olarak bir tesis, birtakım farklı parça tiplerini ürettiğinde veya çok amaçlı bir makine bazı görev karışımlarını üstlendiğinde rastlanır [53]. Hazırlık süresi, genellikle işlerin sırasına bağlı olmadığı varsayılarak işlem süresinin içinde yer alır. Bununla birlikte, henüz tamamlanan işin tipi, işlenecek olan iş kadar makinenin hazırlık süresini etkileyebilir. Örneğin, toz astarlama üretim hattında ürünler değişik renklerle boyanır. Böyle bir durumda koyu renkli iş yığını astarladıktan sonra boya kutusunu temizlemek için gereken süre açık renk için gerekenden daha fazladır. Sıraya-bağımlı hazırlık sürelerini içeren problemler üzerine yapılan çalışmaların çoğu son işin tamamlanma zamanının en küçüklenmesi gibi daha basit amaç fonksiyonlarını dikkate almıştır. Tek makinede son işin tamamlanma zamanının en küçüklenmesi için bile problem NP-zordur [1]. Bundan dolayı bu araştırmada dikkate alınan E/G problemi NP-zordur.

Çizelge 3.1’de Baker’dan [58] alınan hazırlık süreleri matrisini gösteren bir örnek yer almaktadır. Hazırlık süresi, S_{ij} , i işini takip eden j işi için makinenin hazırlanması için gerekli süredir. Örneğin, 1. işten sonra 2. işin yapılması halinde makinenin hazırlanması için 30 dakika gerekirken, 1. iş 2. işten sonra yapılırsa makinenin hazırlanması için 40 dakika gerekmektedir.

Çizelge 3.1. Sıraya-bağımlı hazırlık süreleri örneği

	S_{ij}			
İş	1	2	3	4
1	-	30	50	90
2	40	-	20	80
3	30	30	-	60
4	20	15	10	-

3.1. Hazırlık Sürelerinin Yarattığı Zorluklar

Sıraya-bağımlı hazırlık süreleri içeren algoritmaları ve sezgiselleri geliştirirken karşılaşılan temel zorluk, çözümün sadece işlerin karakteristiklerine bağlı olmayıp aynı zamanda işler arasındaki etkileşime de bağlı olmasından kaynaklanır. Örneğin, E/G probleminde eğer hazırlık süreleri yer almıyorsa, bir iş p_j , d_j , α_j ve β_j ile tanımlanabilir. Oysa probleme hazırlık süreleri katılırsa S_{ij} faktörü de göz önüne alınmalıdır. Hazırlık süreleri içermeyen E/G problemi üzerine yapılan çalışmaların çoğu basit prensiplere dayanır. Mesela, j işinin tamamlanma zamanı $C_j = p_j + C_{j-1}$, $C_{j-1} = p_{j-1} + C_{j-2}$, ... şeklinde hesaplanır. Bu aşağıdaki gibi tekrar ifade edilebilir:

$$C_j = \sum_{i=1}^j p_i \quad (3.1)$$

Bu sade gösterim sıraya-bağımlı hazırlık sürelerinin yer aldığı durumlarda kullanılmaz. Bu durumda j işinin tamamlanma zamanını bulmak için $C_j = C_{j-1} + S_{j-1,j} + p_j$ kullanılmalıdır. p_j , sıranın ne olduğuna bakılmaksızın her zaman aynı olurken S_{ij} değeri sıraya bağlı olarak değişir. Bu durum problemin yapısını zorlaştırdığı için çalışma yapmak, algoritma ve sezgisel geliştirmek güç hale gelir.

Probleme daha fazla kısıt ilave eden diğer bir önemli konu ise çizelgenin mümkünlüğüdür. Farz edelim ki çizelgeleyici en küçük hazırlık süreli işle başlamak istiyor ve onu sıranın ilk pozisyonuna koyuyor. Böyle bir karar verildiğinde başka bir iş birinciye takip etmek zorunda kalır çünkü bu karar ikisi arasındaki hazırlık süresine dayalı olarak verilmiştir. Eğer prosedür sonraki en küçük hazırlık süreli işin seçimi ile devam ederse başka bir işi takip etmek için daha önce atanmış bir işin yeniden çizelgelenmesiyle muhtemelen mümkün olmayan bir çizelge oluşturulacaktır [3].

Hazırlık süreleri içeren problemler yeterince araştırılmadığı için ortak teslim tarihli E/G probleminin kısıtlandırılmış ve kısıtlandırılmamış durumları arasında kesin bir ayrım bulunmamaktadır. Bilindiği üzere, alt sınır, en iyi çözümlerin olmadığı

durumlarda algoritmaların ve sezgisellerin performansını test etmek için kullanılır. Ancak bu problem için bilinen bir alt sınır yoktur. Morton ve Pentico'ya [69] göre hazırlık maliyetleri çizelgeleme problemleri için genellikle kesin bir alt sınır bulmayı zorlaştırır.

Az sayıdaki iş için en iyi çözümler bulan 0/1 KTP ve dinamik programların formüle edilmesi dışında sıraya-bağımlı hazırlık süreleri içeren E/G problemi hakkında az sayıda çalışma yapılmıştır. Bu nedenle bu çalışmada genel olarak problemin temel modeli ele alınmıştır. Yani tek makine, ortak teslim tarihi ve eşit E/G cezaları söz konusudur. Ancak teslim tarihinin kısıtlandırılmış halinin incelenmesi ve sıraya-bağımlı hazırlık sürelerinin yer alması ile literatüre katkı sağlanmıştır. Amaç fonksiyon ise $f(S) = \sum_{j=1}^n (E_j + T_j)$ olarak kalır.

3.2. Teoremler ve Özellikler

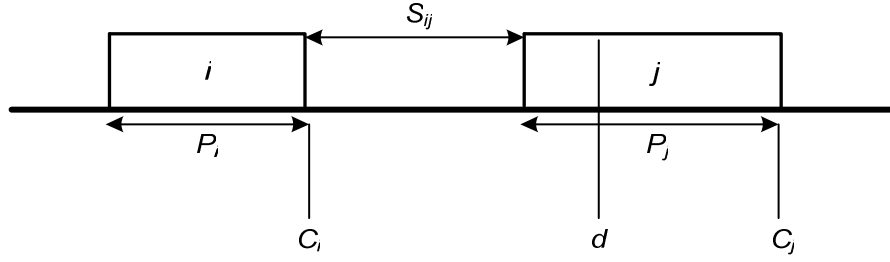
Hazırlık süreleri sıraya-bağımlı olduğunda literatürde E/G problemi için algoritma ve sezgisel geliştirmeye yardımcı olacak teorem ve özellik yoktur. Sonuç olarak, bu bölümde kısıtlandırılmış ortak teslim tarihli problemlerde sıraya-bağımlı hazırlık sürelerinin ortaya çıkması durumunda teoremlerin hala sağlanıp sağlanmadığı incelenmiştir [3].

Teorem 3.1: Çizelgede boş zaman yer almaz.

Bu teorem boş zaman içeren ve içermeyen iki çizelgenin karşılaştırılmasıyla kolayca ispatlanabilir. İçermeyen çizelgede işler arasında boşluk kalmadığından işlerin tamamlanma zamanları d' 'ye daha yakın olur ve böylece çizelgenin maliyeti azalır, bu da onu daha iyi bir çizelge yapar. Bu nedenle, bu teori hazırlık sürelerinin ortaya çıkması durumunda sağlanır.

Teorem 3.2: En iyi çizelgede bir iş teslim tarihinden önce başlayıp teslim tarihinden sonra tamamlanabilir. Şekil 3.1'de j işi, i işinden sonra gelmekte, S_{ij} kadar hazırlık

süresinin ardından işlem görmeye başlamakta ve ortak teslim tarihinden sonra tamamlanmaktadır. Kısıtlandırılmış ortak teslim tarihi söz konusu olduğunda bu şekildeki gibi bir çizelge en iyi çözüm olabilir.



Şekil 3.1. Teslim tarihinde tamamlanmayan bir işin gösterimi

Teorem 3.3: E/G problemi literatüründeki en önemli teoremlerden biri de erken işlerin (B kümesi) işlem sürelerinin azalan sırasına göre ve geç işlerin (A kümesi) işlem sürelerinin artan sırasına göre sıralanmasıdır. Bu, V-biçimli çizelge olarak bilinir. Bu teoremin hazırlık sürelerinin olduğu durumlar için sağlanması gerekli değildir. Özellikle hazırlık süreleri işlem sürelerine kıyasla daha büyük olduğunda bu oldukça açıktır. Bu durumda, en iyi çizelge işlem sürelerinden daha çok hazırlık sürelerine bağlı olacaktır. Dolayısıyla sıralama kararı hazırlık ve işlem sürelerinin her ikisine birden dayalı olmalıdır. Bunu sağlamak için de hazırlık ve işlem süreleri bir matriste birleştirilerek *Düzenlenmiş İşlem Süreleri* (AP_{ij}) *Matrisi* elde edilir.

Örnek 3.1: Çizelge 3.2’de 5 iş, işlem ve hazırlık süresi ile birlikte gösterilmiştir.

Çizelge 3.2. Örnek 3.1 için işlem ve hazırlık süreleri

S_{ij}	1	2	3	4	5
1	0	3	7	2	10
2	36	0	9	22	7
3	44	14	0	13	21
4	37	1	6	0	8
5	29	16	2	15	0
P_j	40	90	20	17	80

Çizelge 3.3. Örnek 3.1 için düzenlenmiş işlem süreleri

AP_{ij}	1	2	3	4	5
1	40	93	27	19	90
2	76	90	29	39	87
3	84	104	20	30	101
4	77	91	26	17	88
5	69	106	22	32	80

Burada dikkat edilmesi gereken iki önemli nokta vardır. Bunlardan ilki sıradaki ilk işin hazırlık süresinin sıfır olarak varsayıldığı ve V-biçim özelliğinin sağlanmadığıdır. E/G problemi için çok önemli olan V-biçim özelliği hazırlık süreleri söz konusu olduğunda sağlanmadığı için en iyiyi garanti eden etkili bir algoritma geliştirmek kolay değildir. Bu nedenle etkili sonuçlar elde edebilmek için Bölüm 4'te yeni bir sezgisel geliştirilmiştir.

3.3. Hazırlık ve İşlem Süreleri için Matematiksel Eşitlik

Örnek 3.1'de işlem süreleri hazırlık süresi matrisiyle birleştirilmiştir. Yeni matris, AP_{ij} matrisi olarak adlandırılmıştır. Bunun nedeni kullanımının daha kolay olmasıdır. Sonraki bölümlerde hem işlem sürelerini hem de hazırlık sürelerini tek bir matriste gösteren bu terim sıkça kullanılacaktır. AP_{ij} matrisi matematiksel olarak aşağıda gösterildiği gibi oluşturulmuştur:

$$AP_{ij} = S_{ij} + P_j \quad j = 1, \dots, n \quad i = 1, \dots, n \quad (3.2)$$

S_{ij} matrisinin sütunlarının birindeki en küçük değeri bulmak AP_{ij} matrisindeki aynı sütunun en küçük değerini bulmaya eşittir. AP_{ij} kavramı, hazırlık sürelerinin işlem sürelerinden ayrı olarak ele alındığı tamsayılı programlama formülasyonunun sunulduğu kısım dışındaki bölümlerde kullanılacaktır.

Bölüm 2’de toplam erkenlik ve geçliğin amaç fonksiyonu formüle edilmişti, aynı yaklaşım hazırlık sürelerinin varlığını yansıtan AP_{ij} kavramını içeren probleme uygulanabilir. Böylece toplam erkenlik ve geçlik Eş. 3.3 ve Eş. 3.4 ile yeniden ifade edilebilir:

$$\begin{aligned} \text{Toplam Erkenlik} &= \sum_{j=1}^b (j-1)AP_{[j-1][j]} \\ &= 0AP_{[0][1]} + 1AP_{[1][2]} + 2AP_{[2][3]} + \dots + (b-1)AP_{[b-1][b]} \end{aligned} \quad (3.3)$$

$$\begin{aligned} \text{Toplam Geçlik} &= \sum_{j=b}^{n-1} (n-j)AP_{[j][j+1]} \\ &= 1AP_{[n-1][n]} + 2AP_{[n-2][n-1]} + \dots + (n-b-1)AP_{[n-b-1][n-b]} + (n-b)AP_{[b][b+1]} \end{aligned} \quad (3.4)$$

Burada $[j]$, j pozisyonundaki işi gösterir, j işinin kendi değeridir.

4. GELİŞTİRİLEN ÇÖZÜM YÖNTEMLERİ

4.1. KTP Kullanarak E/G Problemleri için En İyi Çözümler Bulma

KTP, çizelgeleme ve sıralama problemleri gibi kombinatorial problemlerin en iyi çözümlerini bulmak için kullanılan yaklaşımlardan biridir. İkili (0/1) KTP özellikle çizelgeleme problemlerinde yaygın olarak kullanılır çünkü Doğru/Yanlış koşullarını gösterebilmektedir. Örneğin, x_{ij} , ikili bir değişken ve değeri 1 ise bu i işinin j pozisyonuna atandığı anlamına gelir; aksi takdirde değeri 0 olacaktır. Kombinatorial problemleri çözmek için KTP'leri kullanmanın başlıca dezavantajı, orta boyutlu problemlerde bile gereken işlemsel sürenin uzunluğudur. Diğer taraftan, KTP'ler en iyi çözümü bulma avantajına sahiptir. Bu nedenle araştırma problemlerinde diğer yöntemlerle karşılaştırma yapmak amacıyla kullanılır. Bu çalışmada KTP, geliştirilen sezgisel ve bu sezgiseli iyileştirmek amacıyla oluşturulan TA algoritması tarafından elde edilen çözümleri değerlendirmek amacıyla, küçük sayıdaki problemler için en iyi çözümleri bulmakta kullanılmıştır.

4.1.1. Hazırlık süreleri üçgensel olan E/G problemi için KTP

Sıraya-bağımlılığın en genel halinde S_{ij} ve S_{ji} keyfi değerlere sahiptir. Bu durumda, hazırlık süreleri herhangi bir $n \times n$ matrisiyle gösterilebilir. Bazı durumlarda ise Eş. 4.1'de verilen üçgensel eşitsizlik uygulanarak sıraya-bağımlı hazırlık sürelerinin özel bir hali elde edilir [3]:

$$S_{ij} + S_{jk} \geq S_{ik} \quad \forall \quad 1 \leq i, j, k \leq n \quad (4.1)$$

Tek-makineli E/G problemi için en iyi çözümler bulmak amacıyla Coleman [61] aşağıdaki koşullar altında problemi çözen bir 0/1 KTP formüle etmiştir:

- 1) Farklı teslim tarihleri
- 2) İşe-bağımlı E/G cezaları

3) Eş. 4.1 ile gösterilen üçgensel eşitsizliği sağlayan sıraya-bağımlı hazırlık süreleri

Coleman [61] tarafından geliştirilen model, bazı değişiklikler yapılarak bu çalışmadaki E/G problemine uygun hale getirilmiştir. Bu değişikliklerle kastedilen farklı teslim tarihlerinin ortak bir teslim tarihine ve işe-bağımlı cezaların eşit cezalara dönüştürülmesidir. Düzenlemeden sonra model aşağıdaki halini alır [3]:

$$\text{Min } Z = \sum_{i=1}^n (E_i + T_i) \quad (4.2)$$

Kısıtlar

$$C_i \geq P_i + S_{ij} \quad (4.3)$$

$$C_i - T_i + E_i = d \quad (4.4)$$

$$C_j - C_i + M(1 - Y_{ij}) \geq P_j + S_{ij} \quad i = 1, \dots, n \quad j = i + 1, \dots, n \quad (4.5)$$

$$C_i - C_j + M(Y_{ij}) \geq P_i + S_{ji} \quad i = 1, \dots, n \quad j = i + 1, \dots, n \quad (4.6)$$

$$C_i, E_i, T_i \geq 0 \quad (4.7)$$

$$Y_{ij} \in \{0,1\} \quad (4.8)$$

Burada,

C_j : j işinin tamamlanma zamanı

P_j : j işinin işlem süresi

S_{ij} : i işinden sonra gelen j işi için gereken hazırlık süresi

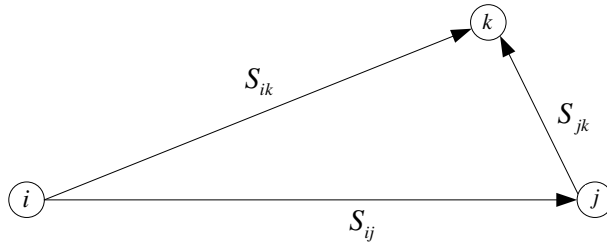
Y_{ij} : i işinden sonra j işi gelirse 1, aksi takdirde 0

M : Büyük bir pozitif sayıdır.

E/G cezalarının eşit olduğu varsayılırsa amaç fonksiyonu Eş. 4.2 ile gösterilir. Eş. 4.3, çizelgenin sıfır zamanından önce başlamadığını garanti eder. Çizelgedeki ilk işin bir hazırlık süresi gerektirdiği durumlar için Eş. 4.3'te S_{ii} yer alır. Coleman [61] deneylerinde sistemin mevcut durumu için hazırlık süresinin gerekmediği varsayımını kullanmış ve böylece $S_{ii} = 0$ olmuştur. Bu çalışmada da $S_{ii} = 0$ olarak alınmıştır. Eş. 4.4, d 'ye göre erken ve geç işlerin periyotlarının sayısını belirleyen

doğrusal programlamanın negatif olmama özelliğini sağlar. Eş. 4.5 ve Eş. 4.6, hangi işten sonra diğerinin geleceğine karar veren Y_{ij} değişkenini kullanarak işlerin tamamlanma zamanlarını bulan ayrık kısıtlardır. M , büyük bir tamsayı olduğu için Eş. 4.5'in, j 'nin i 'yi izlediği durumda aktif ve Eş. 4.6'nın gereksiz olacağı belirtilmelidir. i, j 'yi izlediğinde ise tersi doğru olur. Eş. 4.7 negatif olmama kısıtıdır ve son olarak Eş. 4.8, Y_{ij} değişkenlerinin ikili olduğunu gösterir. Hazırlık süreleri, S_{ij} , üçgensel olmalı yani Eş. 4.1'i sağlamalıdır.

Sıraya-bağımlı hazırlık süreleri içeren problemlerle ilgili GSP'yi anlamak önemlidir. GSP'de bir satıcı n sayıda şehri gezmek zorundadır. Her şehir sadece bir kez ziyaret edilmelidir. Satıcı gidilecek uzaklığı en küçükleme ister. i ve j şehirleri arasındaki uzaklık i işinden sonraki j işi için gereken hazırlık süresine bağlıdır. Herhangi üç şehir arasındaki uzaklık bir üçgen oluşturur ve herhangi iki şehir arasındaki direkt mesafe üçüncü şehirden gidildiği takdirdeki mesafeden büyük olamaz (Şekil 4.1). E/G problemi açısından, makineyi iki kere hazırlamak çoğu zaman bir kere hazırlamaktan daha uzun süreceği için GSP'ye dayanan üçgensel eşitsizlik varsayımı bu çalışmada da kullanılmıştır.



Şekil 4.1. Üçgensel eşitsizlik

4.1.2. Modelin kurulumu

Yukarıda anlatılan KTP modelini kurmak için pek çok en iyileme yazılım paketi kullanılabilir. Doğrusal, doğrusal olmayan ve tamsayılı programlama yazılımı genellikle temel iki bölümden oluşur: Bir modelleme dili ve bir çözücü. AMPL, LINGO, MPL ve MS-Excel gibi modelleme dilleri, matematiksel modelleri geliştirmede kullanıcılara yardımcı olur. CPLEX, LINDO, LP_SOLVE, MINOS ve

MS-Solver gibi çözücüler belli bir model için en iyi çözümler elde edebilecek algoritmaları çalıştırır. Farklı paketler farklı özellik, yetenek ve fiyatlara sahiptir. E/G problemini mümkün olduğu kadar etkili çözmek için birtakım modelleme dilleri ve çözücüler araştırılmıştır. E/G modeli AMPL kullanılarak formüle edilmiştir. Bu dilde yazılmış model Ek-1'dedir. LINGO'nun kullandığı çözücü LINDO iken AMPL, CPLEX, MINOS, LP_SOLVE gibi çözücülerle birleştirilebilir. LINDO 4.0, CPLEX 8.0 ve LP_SOLVE 3.0 kullanılarak yapılan birkaç deney sonucunda CPLEX 8.0'ın en etkilisi olduğu görülmüştür. Bu nedenle, en iyi çözümler bulmak için CPLEX 8.0 ile birlikte modelleme dili olarak AMPL kullanılmıştır.

4.1.3. Veri üretimi

Veriler Balakrishnan ve arkadaşları [70] tarafından kullanılanlara benzer istatistiksel dağılımlardan üretilmiştir. İşlem süreleri, P_j , Düzgün~(0, 100] dağılımdan, sıraya-bağımlı hazırlık süreleri, S_{ij} , Düzgün~(0, 50] dağılımdan rassal olarak üretilmiştir. Teslim tarihi tüm işler için ortaktır ve kısıtlandırılmıştır. Ortak teslim tarihi kısıtlayıcı faktör ve işlem sürelerinin toplamının çarpımıyla elde edilir ($d = \left\lceil h \sum_{j=1}^n p_j \right\rceil$). Bu faktör çizelgenin başlangıcında üretim hattının ne derece sıkıştırıldığını gösterir [31]. Hino ve arkadaşları [31], Biskup ve Feldmann [32] sıraya-bağımlı hazırlık süreleri içermeyen problemlerinde ortak teslim tarihini bu şekilde kısıtladılmışlardır. Bu çalışmada da aynı faktör kullanılmıştır. Ancak bu çalışmada incelenen problemde sıraya-bağımlı hazırlık süreleri de yer aldığı için teslim tarihi daha fazla kısıtlandırılmış olmuştur. Hazırlık süreleri rassal olarak üretildiği için üçgensel eşitsizliği sağlamak için düzeltilmiştir. Bu amaçla Floyd-Warshall (F-W) algoritması kullanılmıştır. F-W algoritması yönlü bir grafikteki düğümlerin tüm parçaları arasındaki en kısa yolu bulan bir algoritmadır. Tüm düğüm çiftleri arasındaki en kısa yolu bulan diğer bir algoritma Dijkstra'nın algoritmasıdır ancak F-W algoritması daha etkilidir. İşlem ve hazırlık sürelerinin üretiminde F-W algoritmasından yararlanılmıştır. Bu algoritmanın sahte kodu aşağıda verilmiştir. Problemler için 5 ila 11 arasında değişen işlerden oluşan veriler üretilmiştir. 11 işli problemleri çözmek için yaklaşık 40 dakikalık süre gerekli olmuştur. Ancak 12 işli

problemler 10 saat içerisinde çözülememiştir. Bu sonuç 11'den fazla iş için problemi çözmekten kaçınmak gerektiğini göstermiştir. Örnek 4.1'de, bulunan en iyi sonucun yanı sıra CPU süresi ve en iyi sonuç için gereken dal-ve-sınır düğümlerinin sayısı da gösterilmiştir.

Floyd-Warshall algoritması

Girdi: Pozitif girişleri olan bir $n \times n$ $[S'_{ij}]$ matrisi

Çıktı: Üçgensel eşitsizliği sağlayan bir $n \times n$ $[S_{ij}]$ matrisi

Begin

for tüm $i \neq j$ do $S_{ij} = S'_{ij}$;

for $i = 1, \dots, n$ do $S_{ii} = \infty$;

for $j = 1, \dots, n$ do

for $i = 1, \dots, n, i \neq j$ do

for $k = 1, \dots, n, k \neq j$, do

$S_{ij} = \min\{S_{ik}, S_{ij} + S_{jk}\}$

Örnek 4.1: Çizelge 4.1'de 5 işin işlem ve hazırlık süresi verilmiştir. Teslim tarihi $d=99$ ve erkenlik ve geçliğin ağırlığı 1'dir. Amaç, erkenlik ve geçliği en küçükleyen en iyi sırayı bulmaktır.

Çizelge 4.1. Örnek 4.1 için veriler

S_{ij}	1	2	3	4	5
1	0	3	7	2	10
2	36	0	9	22	7
3	44	14	0	13	21
4	37	1	6	0	8
5	29	16	2	15	0
P_j	40	90	20	17	80

Çözüm: AMPL modeli kullanarak amaç fonksiyonun en iyi değeriyle birlikte en iyi tamamlanma zamanları elde edilmiştir. Tamamlanma zamanlarının artan sırasına göre işlerin bulunan en iyi sırası Çizelge 4.2’de gösterilmiştir.

Çizelge 4.2. Örnek 4.1 için en iyi çözüm

İş Sırası	1	3	4	2	5
Tamamlanma Zamanı	42	69	99	190	277
En iyi Değer	356				
Cplex 8.0 CPU (sn)	0,03125				
D&S Düğüm Sayısı	79				

Çizelge 4.3 farklı büyüklükteki problemler için elde edilen en iyi sonuçları özetlemektedir. Her problem büyüklüğü için farklı örnek verileriyle 15 koşturma yapılarak ortalama D&S düğüm sayısı ve CPU süresi elde edilmiştir.

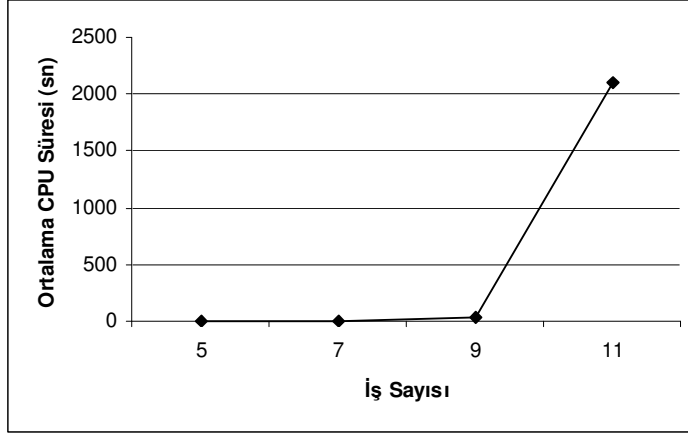
Çizelge 4.3. En iyi çözümlerin özeti

İş Sayısı	Ortalama D&S Düğüm Sayısı	Ortalama CPU (sn)
5	105,20	0,067
7	2386,64	0,789
9	67706,16	30,152
11	3433934,49	2108,957

Elde edilen en iyi çözümler ve işlemsel istatistikler Ek-2’de yer almaktadır. Bu en iyi çözümler, sonraki kısımlarda anlatılan yeni sezgiseli ve bu sezgiseli iyileştirmek için geliştirilen TA algoritmasını değerlendirmek için kullanılacaktır.

Coleman [61] makalesinde 8 işten fazla olan problemlerin uzun bir zaman aldığı sonucuna varmış ve bu nedenle en fazla 8 işle çalışmıştır. Bu tezde ise 11 işe kadarki problemler için en iyi çözümler elde edilmiştir. Bu sonuç geliştirilen algoritmanın test edilebilmesi için daha geniş bir problem çeşitliliği sunar.

Bu tip problemleri çözmek için gereken CPU süresi, iş sayısının artışına bağlı olarak üssel olarak artar. Şekil 4.2, problemin büyüklüğü ile en iyi çözümler bulmak için gereken CPU süresi arasındaki ilişkiyi göstermektedir. Eğer problem büyüklüğü D&S düğümlerine göre çizildiğinde de benzer bir grafik ortaya çıkmaktadır.



Şekil 4.2. İş sayısı ile CPU süresi arasındaki ilişki

Pratikte 11 işli bir problem nispeten küçüktür ve büyük problemler için en iyi çözümler elde etmenin mümkün olup olmadığı merak edilebilir. Bu konuda benzer bir problem olan GSP'den yararlanarak problemin iş sayısının artışına bağlı olarak ne kadar hızlı arttığını bulabiliriz. GSP'de bir satıcı n şehri ziyaret eder. Her şehir ancak ve ancak bir kez ziyaret edilebilir ve kat edilen toplam mesafe en küçüklenmelidir. Dolayısıyla $(n-1)!$ mümkün çözüm vardır. Şehir sayısının 20 ($n=20$) olduğunu ve bir bilgisayarın tüm mümkün çözümleri 1 saat içinde hesaplayabildiğini ($19!=1,2 \times 10^{17}$) varsayalım. O halde 21-şehirlik bir problemi çözmek için 20 saat, 22-şehirlik bir problemi çözmek için 17,5 gün ve 25-şehirlik bir problemi çözmek için 6 yüzyıl gerekir! Bunun nedeni, işlemsel sürenin problemin büyüklüğünün üssel bir fonksiyonu olmasıdır.

4.2. E/G Problemi için Yeni Bir Sezgisel Algoritma

Önceki kısımda görüldüğü gibi en iyi çözümler bulmak amacıyla KTP kullanmak etkili bir yaklaşım değildir. Aslında, böyle metotlar kullanmak orta boyutlu

problemleri en iyi şekilde çözmek için bile mümkün olmayabilir. Çoğu gerçek yaşam problemi büyük olduğu için çabuk elde edilen “iyi” çözümler, hesaplaması saatler alan en iyi çözümlere tercih edilir. Bu nedenle araştırmacılar tatmin edici ve yaklaşık en iyi sonuçlar elde eden zeki araştırma tekniklerini ve sezgiselleri takip etmektedirler.

Basit bir ifadeyle, *algoritma* belirli bir problem için bir çözüm elde edecek adımlar sırasıdır. Algoritmalar üç ana kategoriye ayrılır. İlk kategori *tam doğru algoritmalar*, ikincisi, yaklaşım algoritmaları ve sonuncusu sezgisellerdir. *Tam doğru algoritma* belli bir problemi çözen ve en iyi çözüme ulaştıran algoritmadır. Örneğin Simplex Metodu doğrusal problemleri çözen tam doğru bir algoritmadır. *Yaklaşım algoritması* en iyi çözümü üretmeyen fakat en iyi çözümden belli bir oranda sapmayı garanti eden bir algoritmadır. Son olarak, performansı belirli bir garantiye sahip olmayan algoritmaya *sezgisel* denir.

Bu bölümde, Rabadi’nin [3] kısıtlandırılmamış ortak teslim tarihli E/G problemini çözmek için geliştirdiği sezgisel algoritma kısıtlandırılmış duruma adapte edilecek şekilde önemli bir değişikliğe uğramıştır. Başka bir deyişle, geliştirilen bu yeni sezgiselin algoritma yapısı korunmuş ancak programlama aşamasında oldukça farklılaşmıştır. Bu sezgiselde çizelgeyi oluşturmak için E/G probleminin karakteristikleri kullanılmış ve elde edilen çözüm TA algoritmasına başlangıç çözüm olarak kullanılmıştır.

Sıraya-bağımlı hazırlık süreleri ortaya çıktığında V-biçimlilik özelliğinin sağlanmasına gerek olmadığına Bölüm 3’te değinilmiştir. Bu V-biçim özelliğiyle ilgili olmasına rağmen, en iyi çözümde, uzun işler çizelgenin başlangıcına ve sonuna doğru, kısa işler ortasına doğru çizelgelenme eğilimindedir. *SAPT* (The Shortest Adjusted Processing Time First) sezgiseli de bunu temel alarak düzenlenmiş en küçük işlem süreli işin ilk işlenmesine dayalı bir sezgisel olmuştur.

Çizelgeleme problemleri için geliştirilen çoğu sezgisel iki safhadan oluşur: (1) Çizelge Oluşturma ve (2) Çizelge Geliştirme. İlk safhada genellikle problemin

yapısından kaynaklanan bazı sebeplere dayalı olarak bir çizelge oluşturulur. İkinci safhada, çözümün kalitesini geliştirmek için genellikle bir komşu arama tekniği kullanılır. Aynı prensip *SAPT* sezgiseline uygulanmıştır. Safha 1, AP_{ij} matrisindeki en küçük AP_{ij} değerinin bulunarak i ve j işinin seçimiyle başlar. Sonraki en küçük AP_{kl} 'li iki iş, k ve l , matriste kalan elemanlardan seçilir ve j işinden sonra ya da i işinden önce yerleştirilir, tüm işler çizelgelenene kadar bu böyle devam eder. Çizelgeleme işlemi boyunca olurluğu korumak önemlidir, aksi takdirde uygun olmayan çizelgeler ortaya çıkabilir. Örneğin, j işi i işinden sonra çizelgelenirse, i işi j işinden sonra çizelgelenemez ve bu nedenle algoritma hem AP_{ij} hem de AP_{ji} değerinin matristen çıkarılmasını sağlamalıdır. Diğer olurluk koşulları da hesaba katılmalıdır, bu koşullar izleyen bölümlerde sahte kod olarak sunulacaklardır. Safha 2'de bir komşuluk arama metodu kullanılır. Yerel en iyiyi bulmak için komşuluk arama çok yaygın bir yaklaşımdır.

Komşuluk arama

Komşuluk arama algoritmaları, arama uzayının bir parçasını (komşuluğunu) arayan ve bu komşuluk içinde yerel bir en iyi çözüme ulaşan tekniklerdir. Genellikle iyi bir sezgiselle elde edilen mümkün bir başlangıç çözümü arayarak başlar ve çizelgede mümkün olan (yani mevcut çözümün komşuluğunda kalarak) tüm küçük değişiklik yollarını arar. Sonra her nihai çizelgeyi hesaplar ve eğer hiçbir gelişme olmuyorsa sonlanır. Aksi takdirde, en büyük gelişme sağlayan çözümü alır ve mümkün gelişme olmayana kadar tekrar başlar. Görüldüğü üzere komşuluk arama doğrusal olmayan problemlerde kullanılan “tepe-tırmanma” algoritmalarının özel bir halidir.

Çizelgeleme literatüründe komşuluk aramanın farklı şekilleri bulunmaktadır. Bunlardan en yaygın olanları aşağıdaki gibidir:

1) *Bitişik İkililerin Yer Değiştirmesi* (API): Bir iş, çizelgede sağında veya solunda bulunan işlerle direkt olarak yer değiştirebilir. n iş için aşağıda görüleceği gibi komşuluk $(n-1)$ sıradan oluşur:

$1, 2, 3, 4, \dots, n-2, n-1, n$ (başlangıç çözümü)
 $2, 1, 3, 4, \dots, n-2, n-1, n$ (1 ve 2 yer değiştirir)
 $1, 3, 2, 4, \dots, n-2, n-1, n$ (2 ve 3 yer değiştirir)
 $:$
 $:$
 $1, 2, 3, 4, \dots, n-2, n, n-1$ ($n-1$ ve n yer değiştirir)

Komşuluk aramanın en basit yoludur ve çok etkili olmayabilir.

2) *Yerleştirme*: Her bir iş çizelgedeki başka herhangi bir işin önüne aşağıdaki gibi yerleştirilebilir:

$1, 2, 3, 4, \dots, n-2, n-1, n$ (başlangıç çözümü: 1, 2'nin önünde)
 $2, 1, 3, 4, \dots, n-2, n-1, n$ (1'i 3'ün önüne yerleştirme)
 $2, 3, 1, 4, \dots, n-2, n-1, n$ (1'i 4'ün önüne yerleştirme)
 $:$
 $:$
 $2, 3, 4, \dots, n-2, n-1, 1, n$ (1'i n 'in önüne yerleştirme)

2, 3, ..., n için aynı işlemler uygulanır.

3) *Tüm İkililerin Yer Değiştirmesi (GPI)*: Herhangi iki iş (bitişik olması gerekmez) karşılıklı yer değiştirebilir. Komşuluk, her mümkün ikilinin yer değişimiyle üretilir. n iş için, komşuluk aşağıdaki gibi $n(n-1)/2$ sıradan oluşur:

$1, 2, 3, 4, \dots, n-2, n-1, n$ (başlangıç çözümü)
1 işi için: $2, 1, 3, 4, \dots, n-2, n-1, n$ (1 ve 2 yer değiştirir)
 $3, 2, 1, 4, \dots, n-2, n-1, n$ (1 ve 3 yer değiştirir)
 $:$
 $:$
 $n, 2, 3, 4, \dots, n-2, n-1, 1$ (1 ile n yer değiştirir)

2 işi için: $1, 3, 2, 4, \dots, n-2, n-1, n$ (2 ve 3 yer değiştirir)
 $1, 4, 3, 2, \dots, n-2, n-1, n$ (2 ve 4 yer değiştirir)
 $:$ $:$
 $:$ $:$

 $:$ $1, n, 3, 4, \dots, n-2, n-1, n$ (2 ile n yer değiştirir)
 $:$ $:$
 $:$ $:$
 $:$ $:$

$n-1$ işi için: $1, 2, 3, 4, \dots, n-2, n, n-1$ ($n-1$ ile n yer değiştirir)

SAPT sezgiselinin algoritması

Safha 1. Çizelge Oluşturma

Önceki bölümlerden hatırlarsak:

$B = \text{Erken işlerin sıralanmış kümesi} = \{j : C_j \leq d\}$

$A = \text{Geç işlerin sıralanmış kümesi} = \{j : C_j > d\}$

SAPT sezgiseli en küçük AP_{ij} 'ye sahip i ve j işlerini seçer, i işini d' de tamamlanacak şekilde (b pozisyonuna) yerleştirir, j işi onu izler. Çizelgeyi oluşturmak için üç metot kullanılır: E-Çizelge, G-Çizelge ve EG-Çizelge. En küçük amaç fonksiyonlu çizelge Safha 1'in son çözümü olarak seçilir. Her üç çizelgeyi de oluştururken, sonraki en küçük AP_{kl} 'ye sahip k ve l işleri, çizelgelenmek üzere seçilir. E-Çizelge prosedüründe, önce tüm erken işler, sonra geç işler çizelgelenir. Önce geç işlerin, sonra erken işlerin çizelgelenmesiyle elde edilen sırada G-Çizelgeleme prosedürü izlenmiş olur. EG-Çizelgelemede ise ilk iş çiftinin seçiminden sonra diğer işler eş zamanlı olarak yerleştirilir. Bu kısımda anlatılan bu üç metotla farklı sıralar ve amaç fonksiyonlar elde edilmiş olur. Aşağıda metotlarda kullanılan notasyon bilgileri verilmiştir.

i_min ve j_min : AP_{ij} matrisindeki en küçük elemanla ilgili iki iştir, yani;

$$AP_{i_min, j_min} = \min\{AP_{ij} \mid i \neq j\} \quad \forall i=1, \dots, n, \quad \forall j=1, \dots, n \quad (4.9)$$

i_min^* ve j_min^* : AP_{ij} matrisindeki sonraki en küçük elemanla ilgili iki iştir.

E-Çizelge, G-Çizelge ve EG-Çizelge, E-Çizelge (i_min, j_min), G-Çizelge (i_min, j_min), EG-Çizelge (i_min, j_min) fonksiyonları kullanılarak elde edilir. Bu üç fonksiyon için sahte kod aşağıdaki gibidir [3]:

E-Çizelge (i_min, j_min):

Adım 1. $B = A = \emptyset$, $|B| = b$ ve $|A| = n - b = a$ olsun

Adım 2. i_min 'i B 'deki son boş pozisyona ve j_min 'i A 'daki ilk boş pozisyona ata

Adım 3. AP_{j_min, i_min} 'i sil

Adım 4. $\forall i=1, \dots, n$ için AP_{i, j_min} 'i sil, $\forall j=1, \dots, n$ için $AP_{i_min, j}$ 'yi sil

Adım 5. Erken işleri yerleştir

Adım 5.1. $\forall i=1, \dots, n$ ve $j=i_min$ için i_min^* 'ı bul, öyle ki

$$AP_{i_min^*, j} = \min\{AP_{ij} \mid i \neq j, i \notin A\}$$

Eğer $C_{i_min^*, j} \leq d$ ise Adım 5.2'ye geç

Değilse $|B| = b$ ve $i_min^* \in A$

Adım 5.2. i_min^* 'ı B 'deki son boş pozisyona ata

Adım 5.3. $\forall i=1, \dots, n$ için AP_{ij} 'yi sil, $\forall j=1, \dots, n$ için $AP_{i_min^*, j}$ 'yi sil

Adım 5.4. Eğer $C_{i_min^*, j} \leq d$ ise $i_min = i_min^*$ olur. Adım 5.1'e git

Değilse $|B| = b$ ve Adım 6'ya geç

Adım 6. Geç işleri yerleştir

Adım 6.1. $\forall j=1, \dots, n$ ve $i=j_min$ için j_min^* 'ı bul, öyle ki

$$AP_{i, j_min^*} = \min\{AP_{ij} \mid i \neq j, j \notin B\}$$

Adım 6.2. j_min^* 'ı A 'daki ilk boş pozisyona ata

Adım 6.3. $\forall i = 1, \dots, n$ için AP_{i, j_min^*} 'ı sil, $\forall j = 1, \dots, n$ için AP_{ij} 'yi sil

Adım 6.4. Eğer $|A| < n - b$ ise $j_min = j_min^*$ olur. Adım 6.1'e git
Değilse DUR

Örnek 4.2 : Çizelge 4.4'te $d=114$ olan işlem süreleri ve düzenlenmiş işlem süreleri (AP_{ij}) verilen altı işli ($n=6$) problem için E-Çizelge fonksiyonunun nasıl çalıştığını inceleyelim.

Çizelge 4.4. Örnek 4.2 için işlem ve düzenlenmiş işlem süreleri

AP_{ij}	1	2	3	4	5	6
1	-	21	80	49	24	76
2	23	-	67	61	28	65
3	33	20	-	59	26	77
4	27	28	84	-	32	69
5	29	24	69	44	-	71
6	19	31	76	57	24	-
P	3	8	65	42	11	61

AP_{ij} matrisindeki en küçük değer $AP_{61} = 19$ 'dur. Bu nedenle,

$$i_min = 6$$

$$j_min = 1$$

E-Çizelge (i_min, j_min):

Adım 1. $B = A = \emptyset$, $|B| = b$ ve $|A| = n - b = a$ 'dır.

Adım 2. $B = \{6\}, A = \{1\}$

Adım 3. AP_{16} 'yı sil (1 işi 6 işinden sonra işlem görecektir, bu yüzden 6 işi 1 işinden sonra gelemez)

AP_{ij}	1	2	3	4	5	6
1	-	21	80	49	24	
2	23	-	67	61	28	65
3	33	20	-	59	26	77
4	27	28	84	-	32	69
5	29	24	69	44	-	71
6	19	31	76	57	24	-

Adım 4. 1. sütunu sil (1 işi 6 işinden sonra geldiği için 1 işinden önce başka bir iş gelemez)

6. satırı sil (6 işi 1 işinden önce geldiği için 6 işinden sonra başka bir iş gelemez)

AP_{ij}		2	3	4	5	6
1		21	80	49	24	
2		-	67	61	28	65
3		20	-	59	26	77
4		28	84	-	32	69
5		24	69	44	-	71

Adım 5. Erken işleri yerleştir

Adım 5.1. $j = 6$

$$AP_{i_{\min}^*, 6} = \min\{65, 77, 69, 71\} = 65 \Rightarrow i_{\min}^* = 2$$

$$C_{i_{\min}^*, j} = 73 \leq d = 114 \text{ olduğu için Adım 5.2'ye geç}$$

Adım 5.2. $B = \{2, 6\}$

Adım 5.3. 6. sütunu sil (6 işi 2 işinden sonra geldiği için 6 işinden önce başka bir iş gelemmez)

2. satırı sil (2 işi 6 işinden önce geldiği için 2 işinden sonra başka bir iş gelemmez)

Adım 5.4. $j = 2$

$$AP_{i_{\min}^*, 2} = \min\{21, 20, 28, 24\} = 20 \Rightarrow i_{\min}^* = 3$$

$$C_{i_{\min}^*, j} = 150 > d = 114 \text{ olduğu için } |B| = b = 2 \text{ ve } i_{\min}^* \in A$$

$B = \{2, 6\}$

2. sütunu sil (2 işi ilk iş olduğu için ondan önce başka bir iş gelemmez)

AP_{ij}			3	4	5
1			80	49	24
3			-	59	26
4			84	-	32
5			69	44	-

Adım 6. Geç işleri yerleştir

Adım 6.1. $i = j_{\min} = 1$

$$AP_{1, j_{\min}^*} = \min\{80, 49, 24\} = 24 \Rightarrow j_{\min}^* = 5$$

Adım 6.2. $A = \{1, 5\}$

Adım 6.3. 5. sütunu sil (5 işi 1 işinden sonra geldiği için 5 işinden önce başka bir iş gelemmez)

1. satırı sil (1 işi 5 işinden önce geldiği için 1 işinden sonra başka bir iş gelemmez)

AP_{ij}		3	4		
3		-	59		
4		84	-		
5		69	44		

Adım 6.4. $|A| < n - b = 4$ olduğu için $j_min = 5$ 'tir. Adım 6.1'e dön

Adım 6.1. $i = j_min = 5$

$$AP_{5, j_min^*} = \min \{69, 44\} = 44 \Rightarrow j_min^* = 4$$

Adım 6.2. $A = \{1, 5, 4\}$

Adım 6.3. 4. sütunu sil (4 işi 5 işinden sonra geldiği için 4 işinden önce başka bir iş gelemmez)

5. satırı sil (5 işi 4 işinden önce geldiği için 5 işinden sonra başka bir iş gelemmez)

AP_{ij}		3			
3		-			
4		84			

Adım 6.4. Matriste kalan tek iş 3'tür. O halde $|A| = \{1, 5, 4, 3\}$

Sonuç olarak, oluşturulan çizelge $2 - 6 - 1 - 5 - 4 - 3$ şeklindedir. Erkenlik, geçlik maliyeti ve toplam maliyet Eş. 3.3 ve Eş. 3.4 ile aşağıdaki gibi hesaplanır:

$$\text{Erkenlik Maliyeti} = (2-1) AP_{[12]} = AP_{26} = 65$$

$$\begin{aligned} \text{Geçlik Maliyeti} &= (6-2) AP_{[23]} + (6-3) AP_{[34]} + (6-4) AP_{[45]} - (6-5) AP_{[56]} \\ &= (4 \times AP_{61}) + (3 \times AP_{15}) + (2 \times AP_{54}) + AP_{43} = 320 \end{aligned}$$

$$\text{Toplam Maliyet} = 65 + 320 = 385$$

E-Çizelgeye benzer şekilde G-Çizelge oluşturulur fakat bu sefer önce geç çizelge sonrada erken çizelge kurulur. Başka bir deyişle, G-Çizelge (i_min, j_min) fonksiyonunda Adım 5 ve 6 yer değiştirir.

G-Çizelge (i_min, j_min) :

Adım 1. $B = A = \emptyset$, $|B| = b$ ve $|A| = n - b = a$ olsun

Adım 2. i_min 'i B 'deki son boş pozisyona ve j_min 'i A 'daki ilk boş pozisyona ata

Adım 3. AP_{j_min, i_min} 'i sil

Adım 4. $\forall i = 1, \dots, n$ için AP_{i, j_min} 'i sil, $\forall j = 1, \dots, n$ için $AP_{i_min, j}$ 'yi sil

Adım 5. Geç işleri yerleştir

Adım 5.1. $\forall j = 1, \dots, n$ ve $i = j_min$ için j_min^* 'ı bul, öyle ki

$$AP_{i, j_min^*} = \min\{AP_{ij} \mid i \neq j, j \notin B\}$$

Adım 5.2. j_min^* 'ı A 'daki ilk boş pozisyona ata

Adım 5.3. $\forall i = 1, \dots, n$ için AP_{i, j_min^*} 'ı sil, $\forall j = 1, \dots, n$ için AP_{ij} 'yi sil

Adım 5.4. Eğer $|A| < n - b$ ise $j_min = j_min^*$ olur. Adım 5.1'e git

Adım 6. Erken işleri yerleştir

Adım 6.1. $\forall i = 1, \dots, n$ ve $j = i_min$ için i_min^* 'ı bul, öyle ki

$$AP_{i_min^*, j} = \min\{AP_{ij} \mid i \neq j, i \notin A\}$$

Adım 6.2. i_min^* 'ı B 'deki son boş pozisyona ata

Adım 6.3. $\forall i = 1, \dots, n$ için AP_{ij} 'yi sil, $\forall j = 1, \dots, n$ için $AP_{i_min^*, j}$ 'yi sil

Adım 6.4. Eğer $|B| < b$ ise $i_min = i_min^*$ olur. Adım 6.1'e git

Değilse DUR

EG-Çizelge biraz farklı bir şekilde oluşturulur. $i_{\min}$ ve $j_{\min}$ işleri seçildikten ve ortaya yerleştirildikten sonra, sonraki en küçük AP_{ij} değerleri bulunur. Yani $AP_{i_{\min}^*, j_{\min}}$ ve $AP_{i_{\min}, j_{\min}^*}$ ile ilgili $i_{\min}^*$ ve $j_{\min}^*$ işleri sırasıyla seçilir. Hangisinin önce çizelgeleneceğine ve erken mi yoksa geç mi olacağına karar vermek için $AP_{i_{\min}^*, j_{\min}}$ ve $AP_{i_{\min}, j_{\min}^*}$ karşılaştırılır ve küçük olan seçilir. İlk dört adım E-Çizelge ($i_{\min}, j_{\min}$) ve G-Çizelge ($i_{\min}, j_{\min}$)’de olduğu gibidir. Farklılık, erken ve geç çizelgelerin oluşturulmasında meydana gelir. Sahte kodu aşağıdaki gibidir:

EG-Çizelge ($i_{\min}, j_{\min}$):

Adım 1. $B = A = \emptyset$, $|B| = b$ ve $|A| = n - b = a$ olsun

Adım 2. $i_{\min}$ ’i B ’deki son boş pozisyona ve $j_{\min}$ ’i A ’daki ilk boş pozisyona ata

Adım 3. $AP_{j_{\min}, i_{\min}}$ ’i sil

Adım 4. $\forall i = 1, \dots, n$ için $AP_{i, j_{\min}}$ ’i sil, $\forall j = 1, \dots, n$ için $AP_{i_{\min}, j}$ ’yi sil

Adım 5. Erken ve geç işleri eş zamanlı olarak yerleştir

Adım 5.1. $\forall j = 1, \dots, n$ ve $i = j_{\min}$ için $j_{\min}^*$ ’ı bul, öyle ki

$$AP_{i, j_{\min}^*} = \min\{AP_{ij} \mid i \neq j, j \notin B\}$$

$\forall i = 1, \dots, n$ ve $j = i_{\min}$ için $i_{\min}^*$ ’ı bul, öyle ki

$$AP_{i_{\min}^*, j} = \min\{AP_{ij} \mid i \neq j, i \notin A\}$$

Adım 5.2. Eğer $i_{\min}^* = j_{\min}^*$ ise

Eğer $AP_{i, j_{\min}^*} < AP_{i_{\min}^*, j}$ ise $j_{\min}^*$ ’ı A ’daki ilk boş pozisyona ata

$\forall i = 1, \dots, n$ için $AP_{i, j_{\min}^*}$ ’ı sil, $\forall j = 1, \dots, n$ için AP_{ij} ’yi sil

$\forall i = 1, \dots, n$ ve $j = i_{\min}$ için $i_{\min}^*$ ’ı bul, öyle ki

$$AP_{i_{\min}^*, j} = \min\{AP_{ij} \mid i \neq j, i \notin A\}$$

$i_{\min}^*$ ’ı B ’deki son boş pozisyona ata

$\forall j = 1, \dots, n$ için $AP_{i_{\min}^*, j}$ ’yi sil, $\forall i = 1, \dots, n$ için AP_{ij} ’i sil

Değilse $i_{\min}^*$ 'ı B 'deki son boş pozisyona ata

$\forall j = 1, \dots, n$ için $AP_{i_{\min}^*, j}$ 'yi sil, $\forall i = 1, \dots, n$ için AP_{ij} 'i sil

$\forall j = 1, \dots, n$ ve $i = j_{\min}$ için $j_{\min}^*$ 'ı bul, öyle ki

$$AP_{i, j_{\min}^*} = \min\{AP_{ij} \mid i \neq j, j \notin B\}$$

$j_{\min}^*$ 'ı A 'daki ilk boş pozisyona ata

$\forall i = 1, \dots, n$ için $AP_{i, j_{\min}^*}$ 'ı sil, $\forall j = 1, \dots, n$ için AP_{ij} 'yi sil

Adım 5.4'e git

Değilse Adım 5.3'e git

Adım 5.3. Eğer $AP_{i, j_{\min}^*} < AP_{i_{\min}^*, j}$ ise $j_{\min}^*$ 'ı A 'daki ilk boş pozisyona ata

$\forall i = 1, \dots, n$ için $AP_{i, j_{\min}^*}$ 'ı sil, $\forall j = 1, \dots, n$ için AP_{ij} 'yi sil

$i_{\min}^*$ 'ı B 'deki son boş pozisyona ata

$\forall j = 1, \dots, n$ için $AP_{i_{\min}^*, j}$ 'yi sil, $\forall i = 1, \dots, n$ için AP_{ij} 'i sil

Değilse $i_{\min}^*$ 'ı B 'deki son boş pozisyona ata

$\forall j = 1, \dots, n$ için $AP_{i_{\min}^*, j}$ 'yi sil, $\forall i = 1, \dots, n$ için AP_{ij} 'i sil

$j_{\min}^*$ 'ı A 'daki ilk boş pozisyona ata

$\forall i = 1, \dots, n$ için $AP_{i, j_{\min}^*}$ 'ı sil, $\forall j = 1, \dots, n$ için AP_{ij} 'yi sil

Adım 5.4. Eğer $|A| < n - b$ ve $|B| < b$ ise $j_{\min} = j_{\min}^*$, $i_{\min} = i_{\min}^*$ olur.

Adım 5.1'e git

Değilse ve $|B| = b - 1$ ise $i_{\min} = i_{\min}^*$ olur.

$\forall i = 1, \dots, n$ ve $j = i_{\min}$ için $i_{\min}^*$ 'ı bul, öyle ki

$$AP_{i_{\min}^*, j} = \min\{AP_{ij} \mid i \neq j, i \notin A\}$$

DUR

Değilse DUR

Safha 2. Çizelgeyi Geliştirme

Pek çok çizelgeleme sezgiselinde, oluşturulan çizelgeyi geliştirmek için bir komşuluk arama tekniği uygulanır. *SAPT* sezgiseli için ‘Tüm Bitişik İkililerin Yer Değiştirmesi’ ve ‘Yerleştirme’ metotları denenmiş fakat anlamlı bir gelişme göstermemişlerdir. Diğer yandan, GPI komşuluk yapısı, daha fazla ikilinin yer değiştirmesine olanak tanıdığı için daha ümit vericidir. Bu nedenle, E-Çizelge, G-Çizelge ve EG-Çizelge metotları uygulandıktan sonra her birinden bulunan çizelgelere GPI uygulanmıştır.

Çizelgenin oluşturulmasında AP_{ij} matrisinin en küçük elemanı önemli bir rol oynar. Bunun nedeni, E-Çizelge, G-Çizelge ve EG-Çizelge metotlarının hepsinin de bu elemanın seçimiyle başlamasıdır. Bununla birlikte, bu en küçük değer, ondan biraz daha büyük bir değere göre algoritmanın sonucu açısından daha kötü bir sonuç verebilir ancak AP_{ij} matrisindeki çok bir büyük değer başlangıç noktası olarak seçilmesinin algoritmayı daha iyi bir değere götürmesi muhtemel değildir. Çizelgenin kalitesini artırmak için AP_{ij} matrisindeki en küçük eleman seçilir ve bu elemanla ilgili i ve j işleri algoritmanın başlangıç noktası olur. Daha sonra, E-Çizelge, G-Çizelge ve EG-Çizelge metotları ile üç çizelge oluşturulur ve bunlara GPI uygulanır. AP_{ij} matrisindeki sonraki en küçük eleman seçilir ve aynı işlemler yapılır. Eğer daha iyi bir çizelge elde edilirse, önceki çizelge ile sonraki yer değiştirir ve bu prosedür AP_{ij} matrisindeki aynı değere sahip en küçük tüm elemanlara uygulanır. Başka bir deyişle, *SAPT* sezgiseli AP_{ij} matrisinde aynı değere sahip tüm en küçük elemanlara uygulanır. *SAPT* sezgiselini adımlar halinde özetlersek sahte kodu aşağıdaki gibi olacaktır:

SAPT Sezgiseli

Başla:

Adım 1: AP_{ij} matrisindeki en küçük elemanı/elemanları bul

Adım 2: Adım 1’de elde edilen her bir elemanla ilgili $i_{\min}$ ve $j_{\min}$ ’i bul

Adım 3: Adım 2’de elde edilen i ve j için:

Adım 3.1: E-Çizelge ($i_{\min}$, $j_{\min}$)’yi oluştur

Adım 3.2: G-Çizelge ($i_{\min}$, $j_{\min}$)'yi oluştur

Adım 3.3: EG-Çizelge ($i_{\min}$, $j_{\min}$)'yi oluştur

Adım 4: Adım 3'teki en küçük amaç fonksiyonlu çizelgeleri seç

Adım 5: Adım 4'te elde edilen çizelgelere GPI uygula

Adım 6: Eğer AP_{ij} 'deki en küçük tüm elemanlar incelendiyse

DUR

Değilse Adım 2'ye git

Son

4.3. Tabu Arama Algoritması

TA algoritması, komşuluk arama metotlarına ve aramayı yönlendirecek hafızaların ve stratejilerin farklı tiplerinin kullanımına dayalıdır. Kombinatorial en iyileme problemlerinin çözümü için bir strateji olarak ilk kez Glover [71] tarafından önerilmiştir. Algoritma, kombinatorial en iyileme problemleri için genel bir en iyinin bulunması amacıyla bir arama uzayında bir çözümünden başka bir çözüme hareket edilecek şekilde planlanır.

TA Algoritması, aktif çözüm olarak adlandırılan bir başlangıç çözümüyle başlar. Bu başlangıç çözümünün kalitesi algoritmanın performansı için önemlidir. Çizelgeleme problemlerinde bu çözüm, probleme adapte edilen dağıtım kuralları ile üretilebilir. Sonrasında bu başlangıç çözümünden bir komşu çözümler kümesi üretmek için 'hareket' olarak adlandırılan bir operasyon kullanılır. Bu yeni çözümün kalitesini değerlendirmek için bir değerlendirme fonksiyonu kullanılır. Eğer yeni çözümün değeri şimdiki aktif çözümünden daha iyiye, o zaman bu çözüm, yeni en iyi veya elit çözüm olarak elde tutulur. Algoritma bir durdurma koşulu sağlanıncaya kadar bu aşamalı işlemlerle sürdürülür. Algoritmanın bir döngüye girmemesi için, belli sayıdaki önceki hareketi saklayan bir kısa-dönemli hafıza oluşturulur. Her iterasyonda yerel arama algoritması bu kısa-dönemli hafızada veya yasaklı listede olmayan, komşuluğundaki en iyi çözümü bulur. Bu yasaklı liste 'tabu listesi' olarak adlandırılır ve yakın geçmişte ziyaret edilen çözümlerin özelliklerini içerir. Bir

hareket eğer bu tabu listesindeyse belli sayıdaki iterasyonlar için bu harekete izin verilmez. Liste doluysa listede en uzun zaman kalan hareket çıkartılır.

Bazı durumlarda, tabu hareketinin fonksiyon değeri, bir tabu listesinden hareketlerin çıkması için daha önceden tanımlanmış bir aspirasyon seviyesinden daha büyük bir değer alıyorsa tabu hareketine izin verilir. Bir tabu hareketi için bu durum, algoritma tarafından şimdiye dek bulunan en iyiden daha iyi bir çözüm bulunmasını sağlayabilir.

TA algoritması, kuvvetlendirme stratejilerinin kullanımıyla bir önceki elit çözüme yakın arama uzayını ya da çeşitlendirme stratejilerinin kullanımıyla arama uzayının başka bir ziyaret edilmemiş bölgesini araştırarak şekilde planlanabilir. Bir kuvvetlendirme stratejisinde, şimdiki veya elit çözümünden uzak sonuçlar cezalandırılır. Çeşitlendirme stratejileri ise şimdiki çözüme yakın çözümleri cezalandırır. Kuvvetlendirme ve çeşitlendirme stratejileri için cezaları içeren bir fonksiyon değeri düzenlenir.

Algoritma aşağıdaki durumlardan biri sağlanınca duracak şekilde planlanır [72]:

- 1) İterasyon sayısı izin verilen maksimum iterasyon sayısını geçerse
- 2) Şimdiki çözümü geliştirmeyen iterasyon sayısı önceden belirlenmiş bir sayıdan daha büyükse
- 3) Tüm hareketler tabuysa
- 4) Herhangi bir iterasyon için çözüm üretilmiyorsa ve
- 5) En iyi çözüme ulaşıldığına dair kanıt varsa

Şekil 4.3'te TA algoritmasının genel işleyişi en basit haliyle akış şeması olarak gösterilmiştir.

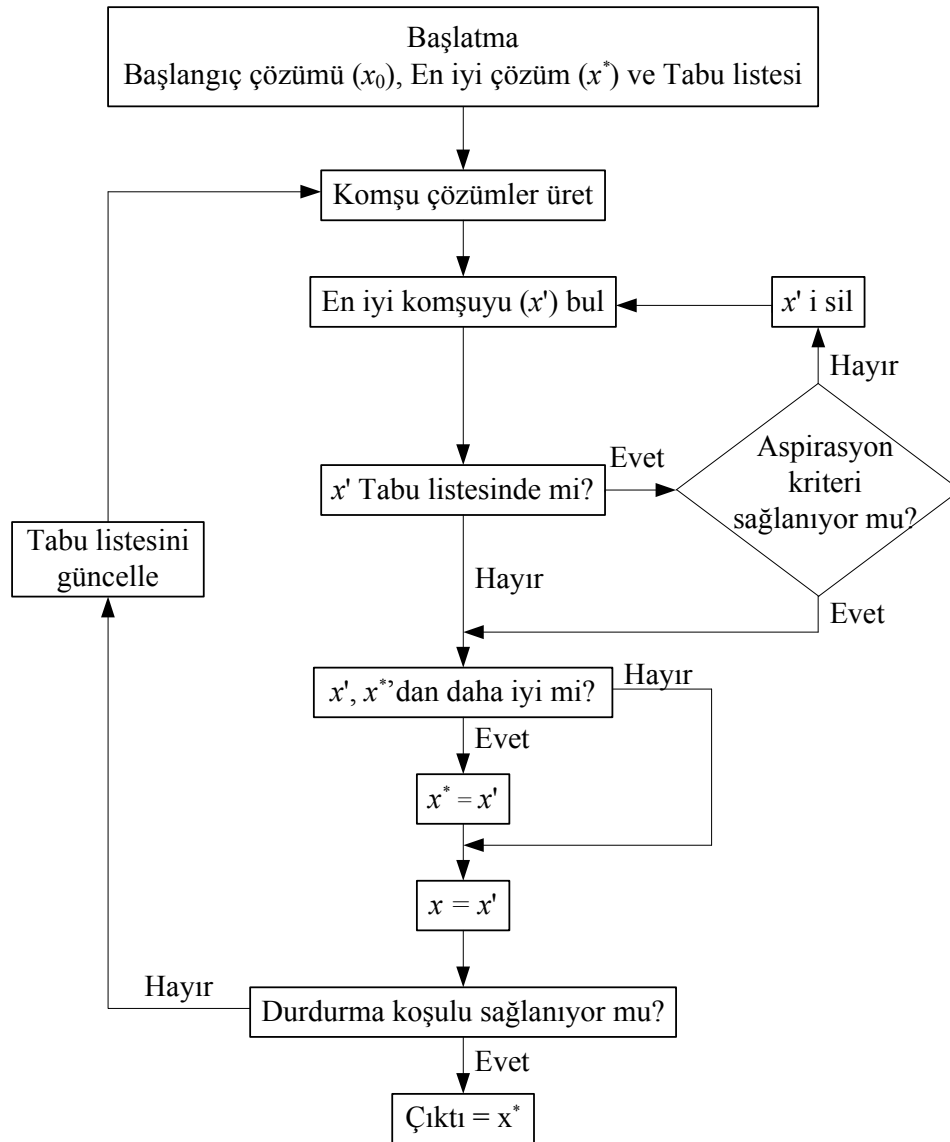
4.3.1. Basit bir TA algoritması

Bu bölümde TA'nın basitleştirilmiş bir hali sunulmuştur. Glover'dan [73] alınan aşağıdaki kavramlar, önerilen basit TA algoritmasını tanımlamak için kullanılmıştır.

Aşağıdaki kombinatorial en iyileme problemini ele alalım:

$$(P) \quad \text{Minimize } c(x) : x \in X \quad R^n$$

Burada $c(x)$, amaç fonksiyonu, x , problem için bir çözümdür ve X mümkün çözümlerin bir kümesinden oluşur.



Şekil 4.3. TA'nın akış şeması

(P) problemini çözmek için planlanan yerel arama prosedürü bir x çözümünden başka bir çözüme $x' \in X$ yönelen hareketlerin bir sırası olarak tanımlanabilir [73]. Burada bir hareket, s , X 'in bir altkümesinden $X(s)$ 'e bir eşlemedir.

$$s: X(s) \rightarrow X$$

Her çözüm için $x \in X$, x 'e uygulanabilen bu hareketler $s \in S$ bir hareketler kümesi $S(x)$ oluşturur:

$$S(x) = \{ s \in S : x \in X(s) \}$$

Benzer şekilde her $s \in S$ için bir hareketle (s) ilgili bir çözümler (x) kümesi vardır:

$$X(s) = \{ x \in X : s \in S(x) \}$$

$x \in X$ olan her çözümün olan bir komşuluğu, $N(x) \subset X$, vardır ve $x' \in N(x)$ olan her çözüm x 'ten bir hareket operasyonu ile elde edilir. Komşuluk şöyle ifade edilir:

$$N(x) = \{ F(x,s) : s \in S(x) \}$$

$F(x,s)$ fonksiyonu ile yeni çözüm, x' , hesaplanır.

Çizelgeleme problemlerinde, x , makinelerdeki operasyonların ve işlerin işlem sırasını gösterir. Basit TA algoritması beş yapısal elemandan oluşur:

- Mümkün bir başlangıç çözümü, x ,
- Bir hareketler kümesi, $S(x)$,
- Bir komşuluk, $N(x)$,
- Bir tabu listesi, T ,
- Bir durdurma koşulu

4.3.2. E/G çizelgeleme probleminde kullanılan TA algoritmaları

Barnes ve Laguna [74], TA kullanılan çizelgeleme literatürü üzerinde inceleme yapmışlardır. Çizelgeleme problemlerini çözmek için uygulanan TA yolları söz edilen problem tipleri kadar çeşitlidir. TA literatüründe bahsedilen bazı çizelgeleme problemlerinin kısa bir özeti aşağıdaki gibi yapılabilir:

1. Akış atölyesi çizelgeleme problemi (Widmer ve Hertz [75], Kim [76], Taillard [77], Reeves [78], Adenso-Diaz [79])
2. İş atölyesi çizelgeleme problemi (Widmer [80], Barnes ve Chambers [81], Sun ve arkadaşları [82])
3. Özdeş paralel makine çizelgeleme problemi (Barnes ve Laguna [83], Hübscher ve Glover [84], Laguna ve González Velarde [85])
4. Hazırlık maliyetleri ve bekleme cezalarının toplamını minimize etme amaçlı tek makine çizelgeleme problemi (Laguna ve arkadaşları [86], Laguna ve Glover [87])

Bu makaleler TA'nın çeşitli çizelgeleme problemlerini çözmedeki uygulanabilirliğini gösterir. James ve Buchanan [88] ise E/G çizelgeleme probleminin özel bir durumundan bahsetmiştir.

James ve Buchanan [89] tarafından E/G çizelgeleme problemlerine TA'nın özel bir uygulaması yapılmıştır. Burada iki farklı TA metodu kullanılarak büyük boyutlu E/G çizelgeleme problemini çözmek için prosedürler tanımlanmıştır. Bir temel oluşturması açısından bu çalışmayı kısaca şöyle özetleyebiliriz: James ve Buchanan [89] tarafından dikkate alınan ilk TA metodu iki-aşamalı bir çizelgeleme ve sıralama yaklaşımı kullanmıştır. Bu metot için çözüm uzayı işlerin tüm farklı sıralarını uzlaştırır ve sıra-tabanlı tabu arama (STTA) metodu olarak adlandırılır. STTA'nın ilk aşamasında işlerin sırası belirlenir ve ikinci aşamasında Fry ve arkadaşlarının [46] boş zaman yerleştirme algoritması kullanılarak işler çizelgelenir. Sonra sonuç çizelgesi maliyetlendirilir. Çizelgedeki işler arasındaki yer değiştirme ve yerleştirme hareketleri komşuluğu tanımlar. STTA metodunda, iki aday seçimi stratejisi kullanılır. Naïve TA olarak adlandırılan ilk stratejide, sonraki hareketi seçmeden

önce tüm aday hareketler incelenir. Bununla birlikte bu strateji mevcut sürede tamamlanan az sayıdaki iterasyonda sonuçlanabilir. Bu, özellikle problemin ve komşuluğun boyutu ile ilgilidir. İkinci strateji Reeves-CL (aday listesi) olarak adlandırılır. TA, Reeves [90] tarafından geliştirilen sadece her iterasyondaki tüm mümkün aday hareketlerin rassal bir altkümesini dikkate alan bir rassal aday listesi stratejisi kullanır. Bu iki stratejili deneyler Reeves [90], Laguna ve arkadaşlarının [86] daha önceki çıkarımlarını destekler. Bu çıkarımlara göre, incelenen komşu çözümlerin sayısını kısıtlayan aday seçimi stratejileri TA'nın performansını artırır. STTA metodunun parçası olarak iki başlama noktası test edilmiştir. Teslim tarihlerine göre işlerin sıralanmasıyla belirlenen ilk başlama noktası en erken teslim tarihi (EDD) olarak adlandırılır. İkinci başlama noktası Ow ve Morton [45] tarafından geliştirilen bitişiklik koşullarını kullanarak EDD başlama noktasını en iyiler, buna da Ow başlama noktası denir.

James ve Buchanan [89] tarafından geliştirilen ikinci TA metodu, erken ve geç işleri tanımlayan bir çözüm uzayına dayalıdır. Özel olarak tasarlanan bir sezgisel, işlerle ilgili bu bilgiyi son çizelgeye dönüştürür. Bu arama 'Erken/Geç Tabu Arama' olarak adlandırılır. Sezgisel 0'lar ve 1'lerden oluşan bir vektörle başlar, buna E/G iş spesifikasyonu denir. Burada 0, teslim tarihinde veya teslim tarihinden önce tamamlanan işleri, 1 ise teslim tarihinde veya teslim tarihinden sonra tamamlanan işleri gösterir. Çizelge 4.5'te E/G problemini çözmede kullanılan bazı TA metotları özetlenmiştir.

Çizelge 4.5. E/G problemini çözmek için kullanılan farklı TA metotları

TA adı	Çözüm uzayı	Aday seçimi	Başlama noktası
Naïve	Sıra tabanlı	Naïve	EDD
Reeves-CL	Sıra tabanlı	Reeves	EDD
Reeves-CL Ow başlangıç	Sıra tabanlı	Reeves	Ow ^a ile EDD
E/G tabu	Erken/geç	Naïve	EDD'ye E/G yaklaşımı
Reeves-CL EG	Erken/geç	Reeves	EDD'ye E/G yaklaşımı
Reeves-CL EG Ow başlangıç	Erken/geç	Reeves	Ow ^a ile EDD'ye E/G yaklaşımı

^a Ow, Ow ve Morton tarafından uyarlanan bitişiklik durumları anlamına gelmektedir.

4.3.3. TA kullanarak çizelgeleme

TA, çeşitli NP-zor problemlerin çözümü için Glover [71, 73, 91] tarafından bir arama tekniği olarak geliştirilmiştir. Glover [73], TA'dan 'meta-sezgisel' olarak söz eder çünkü bir sezgiselin üzerine konulan bir tekniktir ancak kendisi bir sezgisel değildir. TA'ya bağlı olarak kullanılan sezgisel arama için çözüm uzayını ve komşuluğu tanımlar. TA, sonraki çözümü şimdiki çözümün komşularının hepsini veya bir kümesini değerlendirerek seçer ve en iyi amaç fonksiyon değerli komşuya hareket eder. Sezgisel tarafından yapılan hareket 'tabu' olur ve belirli bir iterasyon sayısı geçilene ya da bu hareketin ters çevrilmesiyle elde edilen sonuç çözüme kadar ters çevrilemez. Bir hareketin tabu durumunun iptali için kullanılan aspirasyon kriteri genellikle hareketin yapılmasıyla elde edilen çözümün şimdiye kadar bulunan en iyi çözümden daha iyi olmasıdır [66].

TA çok sayıda çizelgeleme uygulamasında kullanılmasına rağmen hiçbir işe-bağımlı erkenlik ve geçlik cezaları olan kısıtlandırılmış ortak teslim tarihli problemi ele almamıştır (Detaylı bilgi için Bkz. [92]). TA çizelgeleme literatürünün çoğunluğu işlerin fiziksel sırasına dayalı bir çözüm uzayını kullanır. Bu tip çözüm uzayı, böyle

bir kısıtlandırılmış ortak teslim tarihli E/G çizelgeleme problemi ile kullanıldığında, ürettiği çözümlerin çoğu V-biçimliliği sağlamadığı için ikincil en iyi olarak görülür.

Bu ikincil en iyi çözümleri aramaktan kaçınmak ve sadece V-biçimli olan çözümleri incelemek için Hao ve arkadaşları [93] tarafından alternatif bir yaklaşım önerilmiştir. Araştırmacılar, aşağıda ana hatlarıyla gösterilen algoritmayı kullanarak işlerin çizelgesini oluşturmuşlardır. İşlerin çözüm uzaylarının sırasını belirlemek için, $n!$ iş permutasyonu incelenmelidir. İkilik çözüm uzayını sıralamak için 2^n erken/geç iş yerleşimi incelenmelidir. İkilik çözüm uzayı yapısı kullanılarak çözüm uzayı büyüklüğü $n!$ 'den 2^n 'e düşürülür. Bu da n büyüdükçe büyük önem taşır. İkilik çözüm uzayının kullanılması araştırılan tüm çözümlerin daha yüksek kalitede olmasını garanti eder çünkü her zaman teslim tarihinin her iki yanındaki iş sırası açısından daima en iyi olduğu anlamına gelen V-biçimli özelliğe bağlı kalır. Bu çözüm uzayından E/G çözüm uzayı olarak söz edilmiştir.

E/G çözüm uzayı kullanıldığında eğer erken olarak gösterilen tüm işlerin işlem sürelerinin toplamı teslim tarihinden büyükse elde edilen çözümlerden bazıları mümkün olmayacaktır.

Çizelge değiştirme algoritması için E/G iş spesifikasyonu

1. Tüm erken işleri sıranın başlangıcına hareket ettir
 2. Tüm geç işleri sıranın sonuna hareket ettir
 3. Tüm erken işleri p_i / α_i 'nin azalan oranına göre sırala
 4. Tüm geç işleri p_i / β_i 'nin artan oranına göre sırala
 5. Erken olarak tanımlanan işlerin sayısını, b , belirle
 6. Eğer $\sum_{j=1}^b p_j > d$ ise bu E/G iş spesifikasyonu mümkün değildir çünkü tüm erken işlerin tamamlanması için gereken işlem süresi mevcut teslim tarihinden daha fazladır, bu nedenle hata verir.
 7. Tamamlanma zamanını $C_b = d$ belirle. $C_i = C_{i+1} - p_{i+1}$ ilişkisini kullanarak çizelgede geriye kalan tüm işlerin tamamlanma zamanlarını hesapla.
-

4.3.4. Önerilen TA algoritması

TA algoritmasının işleyişi basitçe şöyledir: Bir başlangıç seçimi ele alınır, bu seçimin komşuları bir komşuluk yapısıyla belirlenir, bir amaç fonksiyonuna göre komşular değerlendirilir, değerlendirilen komşu tabu listesinde yoksa ya da aspirasyon kriterini sağlıyorsa bu çözüm şimdiye kadar bulunan en iyi çözümle karşılaştırılır ve tabu listesine eklenir, eğer çözüm en iyi çözümden daha iyi ise sonraki arama için yeni başlangıç çözümü olarak alınır. Bu işlem bir durdurma koşulu karşılanıncaya kadar tekrarlanır. Burada TA ile ilgili bazı kavramlar ön plana çıkmaktadır. Bunlar başlangıç çözümü, komşuluk yapıları, tabu listesinin düzenlenmesi, aspirasyon kriteri ve durdurma koşuludur.

Başlangıç çözümü

TA algoritması herhangi bir mümkün çözümle başlatılmalıdır. Bu, problemin matematiksel ifadesinde geçen kısıtları sağlayacak şekilde işlerin sıralandığı bir çözümdür. Elbette ki TA algoritmasının performansı başlangıç çözümüne oldukça bağlıdır. Bu yüzden TA'ya mümkün olduğunca iyi bir çözümle başlanmalıdır. Başlangıç çözümü rassal ya da bir algoritma kullanarak seçilebilir. Bunların dışında daha önceden iyi olduğu saptanmış bir çözümle de arama başlatılabilir. Bu çalışmada, *SAPT* sezgiseli ile bulunan çözüm, TA'nın başlangıç çözümü olarak kullanılmıştır.

Komşuluk arama ve aday seçimi

Çizelgeleme problemlerine uygulanan TA teknikleri yaygın olarak gezgin satıcı problemlerinde olduğu gibi aynı komşuluk yapısını benimser [91]. Bu çalışmada, API komşuluk yapısı kullanılmıştır. Üretilen $n-1$ adet komşudan en küçük amaç fonksiyon değerine sahip olan komşu, aday çözüm olarak seçilmiştir.

Tabu listesinin düzenlenmesi

Tabu listesi belli bir süre müddetince tekrar göz önüne alınmaması gereken

çözümleri karakterize eden özellikleri saklar. Genellikle listeye ilk-giren-ilk-çıkarm stratejisi uygulanır. Listedeki varlık sayısı liste uzunluğuna ulaştıktan sonra listeye yeni varlıklar tepeden girdikçe eski varlıklar bir aşağı kayar ve en dipteki silinir. Komple bir seçimi bütün olarak tabu listesinde saklamak büyük bir hafıza israfına neden olur. Bu yüzden bu seçimi temsil eden bir özellik listede tutulur. Özellik olarak genellikle en son taşımada ters çevrilen (i,j) faaliyet çifti yayı kullanılır. Dolayısıyla, eğer özellik setine ait bir yay bir S çözümü içinde varsa bu S çözümü tabu olarak tanımlanır. Tabu listesi uzunluğu belirlenmesi gereken önemli parametredir. Liste uzunluğu sabit veya belli bir aralıkta değişken olabilir. En iyi liste uzunluğunu tespit eden kesin bir kural yoktur; deneme yanılma ile tespit edilmeye çalışılır.

Çalışmada, yukarıda anlatılan şekilde Tabu listesi düzenlenmiştir. Liste uzunluğu, belli bir aralıktaki değişkenler olabileceği gibi kullanıcının uygun bulduğu bir uzunluk da olabilir.

Aspirasyon kriteri

TA algoritmasında bir hareketin tabu durumunu iptal etmek için yaygın olarak kullanılan aspirasyon kriterleri şöyledir [30]:

- (1) Eğer bir tabu hareketi şimdiye kadar bulunan en iyi çözümünden daha iyi bir çözüm sağlıyorsa tabu durumu iptal edilir ve harekete izin verilir.
- (2) Eğer tüm hareketler tabuysa ve hiçbirisi aspirasyon kriterini sağlamıyorsa tabu olmayan bir hareket bulunana kadar Tabu listesindeki en eski hareketin tersi seçilir ve “en yeni” hareketin tekrarları listeye eklenir.

Aspirasyon kriteri, bir taşıma üzerindeki tabu durumunun etkisini iptal eden bir koşul olarak tanımlanır. S çözümünden S' çözümüne yasaklanmış bir taşımaya izin vermenin temel koşulu, S' çözüm değerinin şimdiye kadar bulunan en iyi çözüm değerinden daha iyi olmasıdır. Çalışmada da kullanılan bu kriterin arkasındaki fikir şudur; ilgili taşıma tabu olsa bile önceki en iyi çözümü iyileştiren bir taşımayla ilgilenilmelidir.

Durdurma koşulu

Algoritmayı sonlandırmak için yaygın olarak şu şartlar kullanılır:

- En iyi çözümü olduğu bilinen problemlerde en iyi sonucun bulunması
- Önceden belirlenmiş olan iterasyon sayısına ulaşılması: Çalışmada bu durdurma koşulu kullanılmıştır.
- Mevcut en iyi çözümde bir iyileştirme sağlanamaması

5. DENEYSEL BULGULAR VE TARTIŞMA

Bu bölüm, *SAPT* sezgiselinin ve TA algoritmasının performansının değerlendirildiği iki alt bölüme ayrılır. Her iki alt bölümde kullanılan performans ölçütü bu metotlarla bulunan çözümlerin en iyi çözümlerden sapmasıdır. Bu iki metot, 11 işe kadarki problemlerin çözümü için kullanılmıştır çünkü makul süre içerisinde en fazla 11 iş için en iyi sonuç elde edilebilmiştir.

5.1. *SAPT* Sezgiseli İçin İşlemsel Deney

Bu kısımda düzenlenmiş en kısa işlem süreli işin ilk önce sıralanmasına dayalı *SAPT* sezgiseli için yapılan deneysel tasarım değerlendirilmiştir. Sonuçlar istatistiksel olarak analiz edilmiş ve ve bulgular değerlendirilmiştir.

5.1.1. Deneysel tasarım

SAPT sezgisel algoritmasının performansını test etmek için tam faktoriyel bir deney tasarımı dikkate alınmıştır. Deneyde kullanılan faktörler; iş sayısı ve teslim tarihini kısıtlandırmak için kullanılan h faktörüdür.

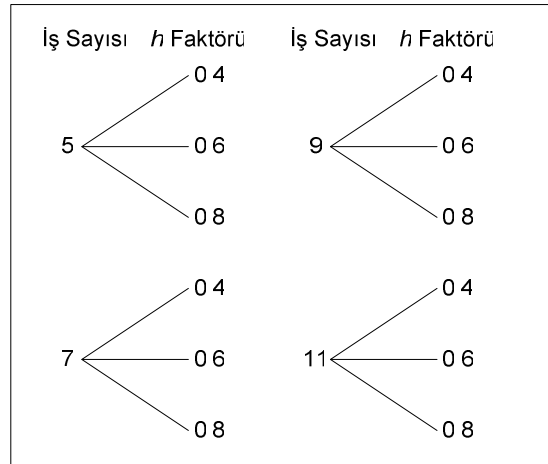
İş sayısı problemin büyüklüğünü belirler, bu nedenle çizelgeleme problemleri için oldukça önemlidir. Bu çalışmada kısıtlandırılmış ortak teslim tarihi ele alındığı için h faktörü de diğer önemli etkidir. Bu iki faktör ve seviyeleri Çizelge 5.1’de gösterilmiştir.

Veriler, rassal olarak örnek setleri üretecek şekilde geliştirilen bir programla üretilmiştir. Çizelge 5.1’den de görüleceği üzere $4 \times 3 = 12$ eşleşme sağlanmıştır.

Çizelge 5.1. *SAPT* deneylerindeki faktörler ve seviyeleri

Faktör	Kısaltma	Seviye
İş Sayısı	Is_Sayisi	5
		7
		9
		11
Kısıtlandırma Faktörü	Kisitlayici_Faktor	0,4
		0,6
		0,8

Bu deneyde, her bir eşleşme durumu için 15 örnek çözülmüştür. Böylece toplam koşturma sayısı $15 \times 12 = 180$ olmuştur.

Şekil 5.1. *SAPT* için tam faktöriyel deney tasarımı

Çizelge 5.2’de tüm koşturmalar sonucunda *SAPT* sezgiseliyle elde edilen sonuçların en iyi sonuçlardan sapmasının değişik açılardan hesapları yer almaktadır. Son sütunda ise *SAPT* sezgiseliyle ulaşılan en iyi sonuç sayısı bulunmaktadır. CPU süreleri tüm problemler için çok küçük değerler aldığı için çizelgede gösterilmemiştir.

Çizelge 5.2. *SAPT* sezgiseli için deney sonuçları

İş Sayısı	h faktörü	En iyiden en küçük sapma (%)	En iyiden ortalama sapma (%)	En iyiden en büyük sapma (%)	Ortalama % sapmanın ortalaması	En iyiye ulaşma sayısı
5	0,4	0	6,58	29,41	4,68	1
	0,6	0	6,16	18,4		1
	0,8	0	1,29	5,53		10
7	0,4	0,34	5,53	11,18	4,13	0
	0,6	0	3,82	12,64		4
	0,8	0	3,05	8,57		3
9	0,4	0,37	6,82	18,48	5,23	0
	0,6	0	4,68	12,55		1
	0,8	0	4,19	9,39		1
11	0,4	2,56	9,36	27,47	8,02	0
	0,6	0,8	6,52	21,64		0
	0,8	2,24	8,19	24,15		0
Ortalama					5,52	

Çizelge 5.2'ye bakıldığında ortalama yüzde sapmanın ortalaması genel olarak iş sayısının artışına bağlı olarak bir artış göstermekte ancak 7 işli deneylerde 5 işli deneylere nazaran daha az sapma olmaktadır. h faktörüne ilişkin de yeterli analiz yapmak mümkün olmadığı için daha istatistiksel bir çalışma yapılmıştır.

5.1.2. İstatistiksel analiz

Bu kısımda, tüm tekrarların sonuçları Çizelge 5.1'de gösterilen faktörler ve seviyeleriyle bir modele uydurulmuştur. Karşılık olarak *SAPT* sezgiseliyle elde edilen sonuçların en iyi sonuçlardan sapma yüzdesi alınmıştır. Verileri analiz etmek ve varyans analizi (ANOVA) çizelgelerini elde etmek için MINITAB istatistik program paketinin 14.12.0 sürümü kullanılmıştır. Tüm model için varyans analizi Çizelge 5.3'te gösterilmiştir. Aşağıda ise deney için kurulan üç hipotez yer almaktadır. Bu hipotezler sırasıyla iş sayısının, h faktörünün ve bu iki faktörün etkileşiminin deney üzerinde etkili olup olmadığının sorgulanmasına yöneliktir.

$$H_0 : \tau_i = 0$$

$$H_1 : \tau_i \neq 0;$$

$$H_0 : \beta_j = 0$$

$$H_1 : \beta_j \neq 0;$$

$$H_0 : (\tau\beta)_{ij} = 0$$

$$H_1 : (\tau\beta)_{ij} \neq 0;$$

Çizelge 5.3. *SAPT* deneyleri için varyans analizi

Değişim Kaynağı	Kareler Toplamı	Serbestlik Derecesi	Ortalama Kare	F ₀	P Değeri
İş Sayısı	403,77	3	134,59	5,47	0,001
Kısıtlayıcı Faktör	255,38	2	127,69	5,19	0,007
Etkileşim	172,19	6	28,70	1,17	0,327
Hata	4135,46	168	24,62		
Toplam	4966,81	179			

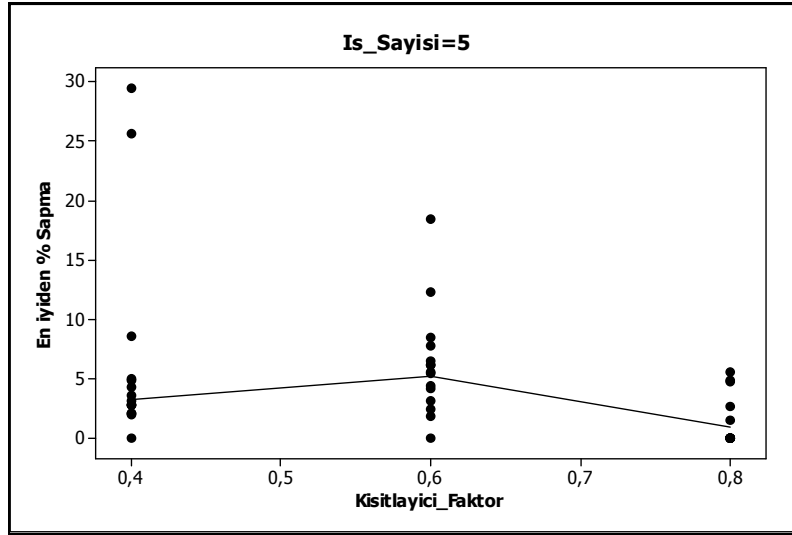
Tüm hipotezler %5 anlamlılık düzeyinde ($\alpha=0,05$) değerlendirildiğinde aşağıdaki sonuçlara varılır:

$F_{\alpha,a-1,ab(n-1)} = F_{0,05,3,168} \approx 2,60 < F_0 = 5,47$ olduğundan H_0 hipotezi reddedilir; iş sayısı deney üzerinde etkilidir.

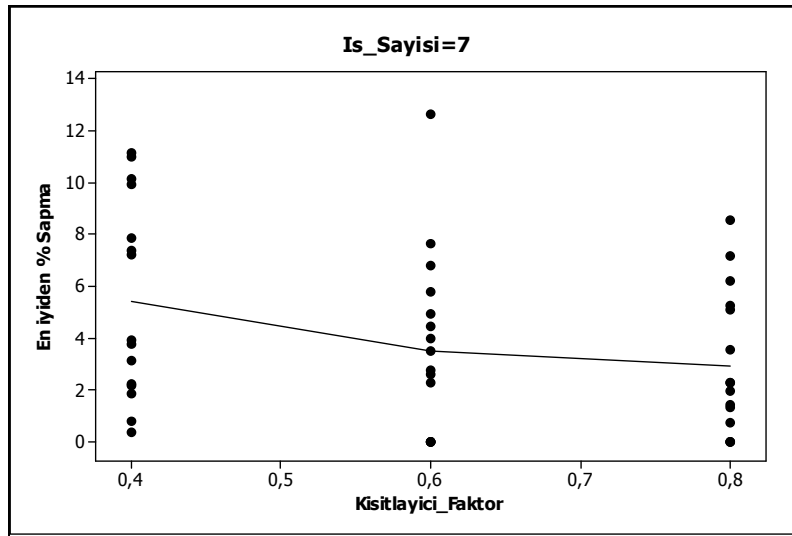
$F_{\alpha,b-1,ab(n-1)} = F_{0,05,2,168} \approx 3,00 < F_0 = 5,19$ olduğundan H_0 hipotezi reddedilir; kısıtlayıcı faktör deney üzerinde etkilidir.

$F_{\alpha,(a-1)(b-1),ab(n-1)} = F_{0,05,6,168} \approx 2,10 > F_0 = 1,17$ olduğundan H_0 hipotezi kabul edilir; iş sayısı ve kısıtlayıcı faktör etkileşimi deney üzerinde etkili değildir.

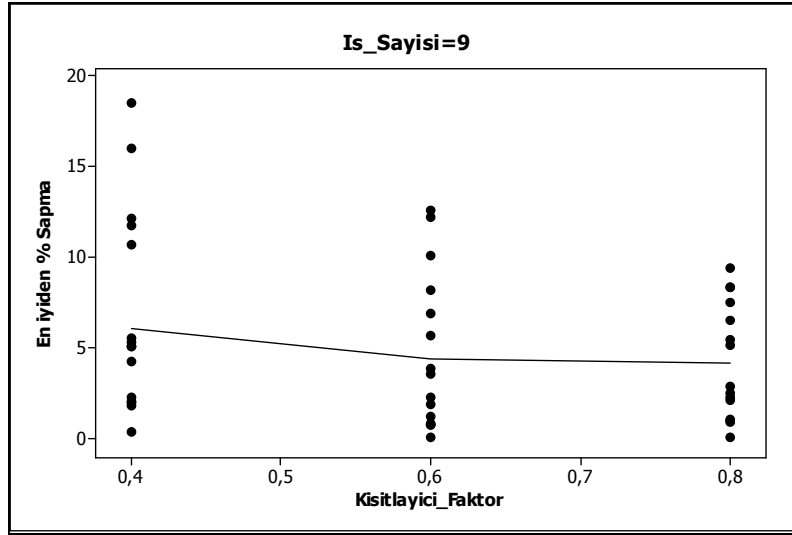
İş sayısı 5, 7, 9 ve 11 olarak sabit tutulduğunda, sırasıyla Şekil 5.2, 5.3, 5.4 ve 5.5 teslim tarihini kısıtlandıran h faktörü ve en iyiden yüzde sapmalar arasındaki ilişkiyi göstermektedir.



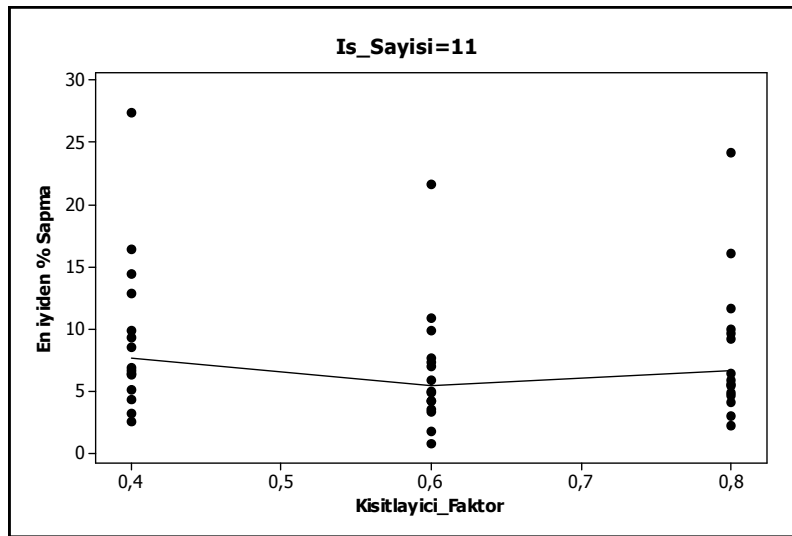
Şekil 5.2. *SAPT* sezgiselinde $n=5$ için h ile en iyiden % sapma arasındaki ilişki



Şekil 5.3. *SAPT* sezgiselinde $n=7$ için h ile en iyiden % sapma arasındaki ilişki



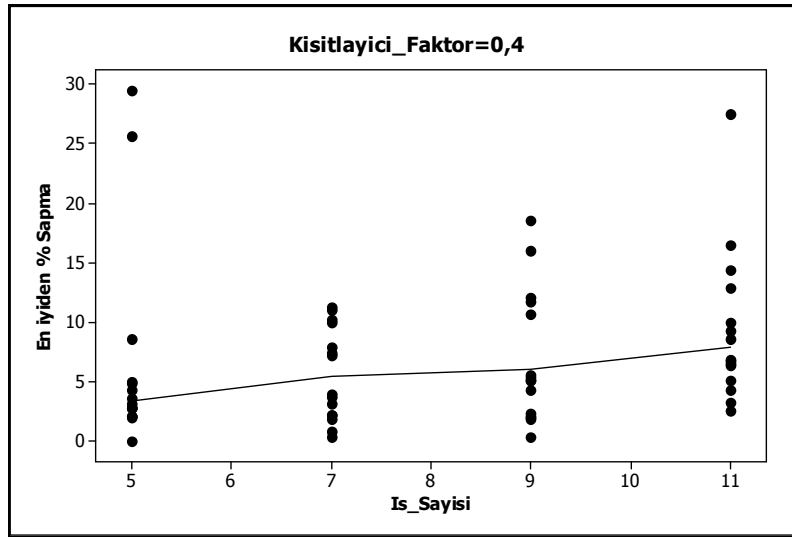
Şekil 5.4. *SAPT* sezgiselinde $n=9$ için h ile en iyiden % sapma arasındaki ilişki



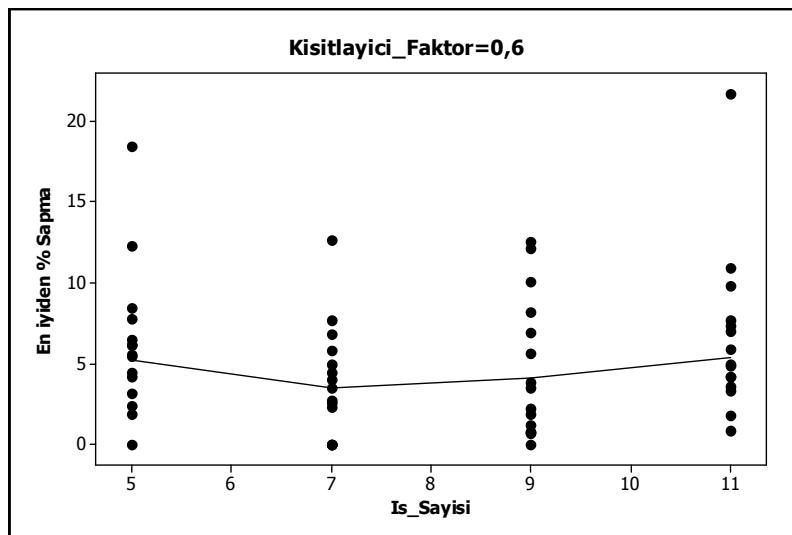
Şekil 5.5. *SAPT* sezgiselinde $n=11$ için h ile en iyiden % sapma arasındaki ilişki

İş sayısı sabit tutulduğunda, teslim tarihini kısıtlandıran faktör en büyük değerini aldığı anda yani $h=0,8$ olduğunda 5, 7 ve 9 işli problemler için en iyiden ortalama yüzde sapma en az olmaktadır. Bunun nedeni oldukça açıktır, h faktörü arttığında teslim tarihi büyür, erkenlik ve geçliği en küçükleyecek amaç fonksiyon değerini sağlayacak iş sırası daha rahat yapılabilir. Böylece en iyiden sapma asgari olacaktır. Ancak 11 işli problem için yapılan deneyler sonucunda h , 0,6 değerini aldığı anda en

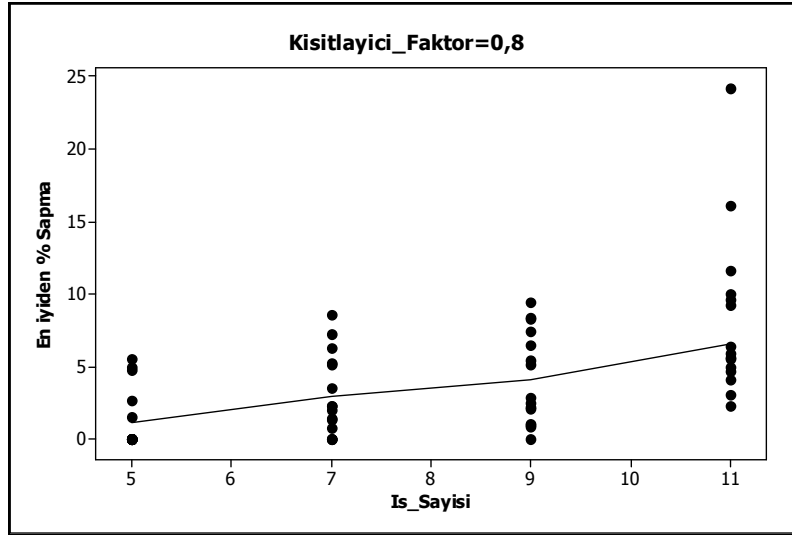
iyiden ortalama yüzde sapmanın en az olduğu görülmektedir. O halde bu konuda genelleme yapamayız. Oysa $h=0,4$ olduğunda en iyiden ortalama yüzde sapmanın en fazla olduğunu rahatlıkla söyleyebiliriz. Benzer şekilde h faktörü sabit tutulduğunda iş sayısı ile en iyiden yüzde sapma arasındaki ilişkiyi gösteren grafikler sırasıyla Şekil 5.6, 5.5 ve 5.8'dir.



Şekil 5.6. SAPT sezgiselinde $h=0,4$ için n ile en iyiden % sapma arasındaki ilişki

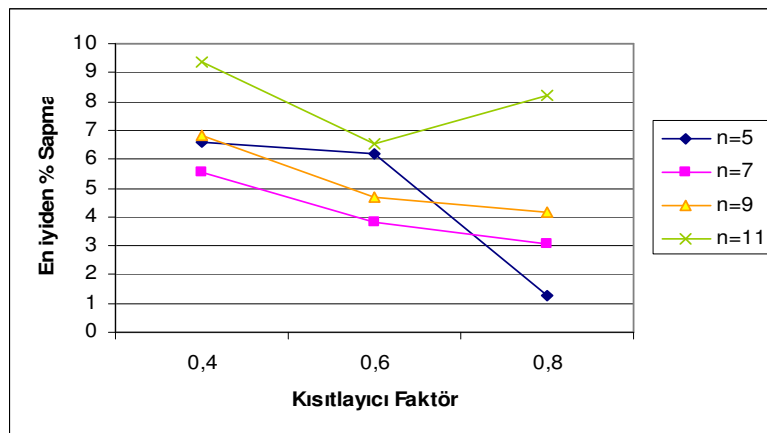


Şekil 5.7. SAPT sezgiselinde $h=0,6$ için n ile en iyiden % sapma arasındaki ilişki



Şekil 5.8. *SAPT* sezgiselinde $h=0,8$ için n ile en iyiden % sapma arasındaki ilişki

Bu sefer grafikleri analiz etmek daha zor çünkü her grafikte farklı eğilimler söz konusudur. En iyiden ortalama yüzde sapmanın en az ve en fazla olduğu durumları ise kısaca özetleyebiliriz. $h=0,4$ ve $h=0,6$ için en iyiden ortalama yüzde sapma 7 iş için en az ve 11 iş için en fazla; $h=0,8$ için en iyiden ortalama yüzde sapma ise 5 iş için en az ve yine 11 iş için en fazladır. Aralardaki değerler ise birbirinden farklı dağılımlara sahiptir. Şekil 5.9'da ise tek bir grafikte iş sayısı faktörünün tüm seviyeleri için kısıtlayıcı faktör ile en iyiden % sapma arasındaki ilişki görülmektedir.



Şekil 5.9. *SAPT* sezgiselinde tüm iş sayıları için h ile en iyiden % sapma arasındaki ilişki

5.2. TA Algoritması İçin İşlemsel Deney

Bu kısımda TA ile bulunan sonuçlar değerlendirilmiş ve en iyi sonuçlarla karşılaştırılmıştır. Ayrıca TA, *SAPT* sezgiselini iyileştirmeye yönelik olarak geliştirildiği için buna ilişkin değerlendirmeler de sunulmuştur.

5.2.1. TA ile elde edilen sonuçlar ve sonuçların analizi

Daha önce *SAPT* sezgiseli tarafından çözülen örnek setleri bu sefer TA kullanılarak çözülmüştür. TA ile bulunan sonuçların en iyi sonuçlardan sapmaları Çizelge 5.4'te gösterilmiştir.

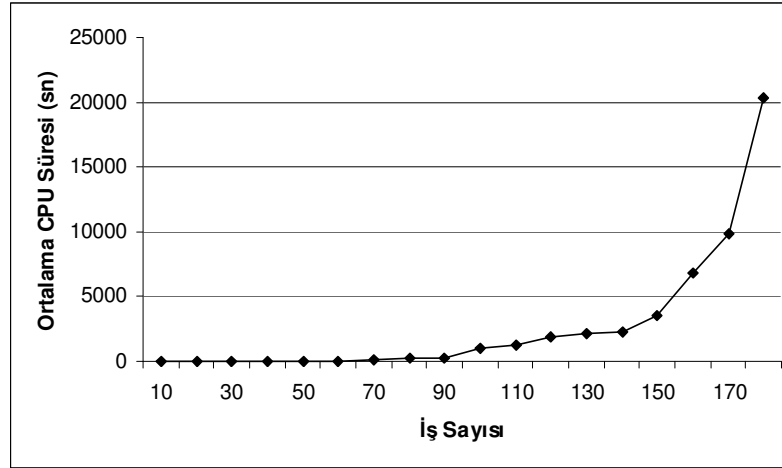
Çizelge 5.4. TA algoritması için deney sonuçları – 1

İş Sayısı	h faktörü	En iyiden en küçük sapma (%)	En iyiden Ortalama sapma (%)	En iyiden en büyük sapma (%)	Ortalama % sapmanın ortalaması	En iyiye ulaşma sayısı
5	0,4	0	4,40	24,63	2,90	2
	0,6	0	3,48	8,71		3
	0,8	0	0,81	4,90		11
7	0,4	0,13	2,62	10,15	1,52	0
	0,6	0	1,27	3,49		5
	0,8	0	0,67	3,80		9
9	0,4	0	1,61	4,56	1,33	1
	0,6	0	0,91	2,96		3
	0,8	0	1,46	6,47		3
11	0,4	0,65	2,73	6,37	2,40	0
	0,6	0	1,42	4,23		3
	0,8	1,03	3,06	6,09		0
Ortalama					2,04	

TA algoritmasını zaman kullanımı açısından değerlendirmek için 10'dan başlamak üzere giderek iş sayısı artırılarak bunun CPU süresi üzerindeki etkisi gözlenmiştir. Çizelge 5.5'te aynı örnek için üç farklı teslim tarihinde elde edilen süreler ve son sütunda ortalama işlem süresi yer almaktadır. Şekil 5.10'dan görülebileceği gibi 180 işten sonra ciddi bir süre gerektirmektedir.

Çizelge 5.5. Farklı problem büyüklüklerinin TA ile çözüm süreleri

İş Sayısı	Koşturma Süreleri (sn)			Ortalama CPU Süresi (sn)
	$h=0,4$	$h=0,6$	$h=0,8$	
10	0,062	0,031	0,015	0,036
20	0,141	0,203	0,187	0,177
30	0,360	0,688	0,515	0,521
40	5,156	1,546	1,500	2,734
50	21,985	9,656	5,157	12,266
60	27,266	28,328	27,859	27,818
70	152,750	31,953	93,141	92,615
80	180,063	239,719	179,984	199,922
90	216,032	322,609	322,735	287,125
100	911,734	897,687	1099,21	969,540
110	1352	1052,65	1228,74	1211,13
120	1905,3	1921,30	1901,10	1909,23
130	1339,5	1980,91	3250,72	2190,37
140	1722,2	858,937	4270,61	2283,91
150	1306,3	4526,10	4830,33	3554,23
160	7344,3	6300,42	6854,21	6832,97
170	10159,3	10075	9254,66	9829,65
180	20037	18500,8	22473,1	20336,97



Şekil 5.10. TA’da iş sayısı ile CPU süresi arasındaki ilişki

5.2.2. İstatistiksel analiz

İş sayısı ve kısıtlandırma faktörü değiştiğinde TA algoritmasının davranışını değerlendirmek için bir varyans analizi (ANOVA) yapılmıştır. Çizelge 5.6 tüm

modelin sonuçlarını göstermektedir. Aşağıda ise deney için kurulan üç hipotez yer almaktadır. Bu hipotezler sırasıyla iş sayısının, h faktörünün ve bu iki faktörün etkileşiminin deney üzerinde etkili olup olmadığının sorgulanmasına yöneliktir.

$$H_0 : \tau_i = 0$$

$$H_1 : \tau_i \neq 0;$$

$$H_0 : \beta_j = 0$$

$$H_1 : \beta_j \neq 0;$$

$$H_0 : (\tau\beta)_{ij} = 0$$

$$H_1 : (\tau\beta)_{ij} \neq 0;$$

Çizelge 5.6. TA algoritmasının 1. deneyi için varyans analizi

Değişim Kaynağı	Kareler Toplamı	Serbestlik Derecesi	Ortalama Kare	F ₀	P Değeri
İş Sayısı	119,35	3	39,78	2,62	0,053
Kısıtlayıcı Faktör	220,33	2	110,17	7,24	0,001
Etkileşim	103,10	6	17,18	1,13	0,347
Hata	2555,69	168	15,21		
Toplam	2998,48	179			

Tüm hipotezler %5 anlamlılık düzeyinde ($\alpha=0,05$) değerlendirildiğinde aşağıdaki sonuçlara varılır:

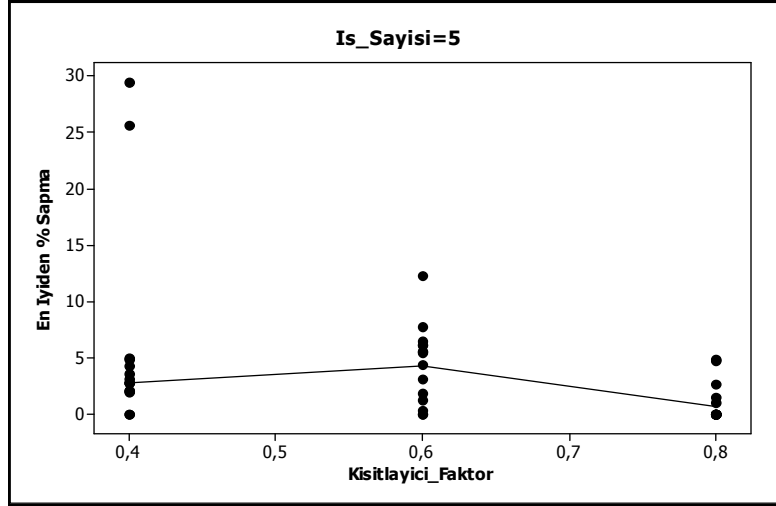
$F_{\alpha,a-1,ab(n-1)} = F_{0,05,3,168} \approx 2,60 < F_0 = 2,62$ olduğundan H_0 hipotezi reddedilir; iş sayısı deney üzerinde etkilidir.

$F_{\alpha,b-1,ab(n-1)} = F_{0,05,2,168} \approx 3,00 < F_0 = 7,24$ olduğundan H_0 hipotezi reddedilir; kısıtlayıcı faktör deney üzerinde etkilidir.

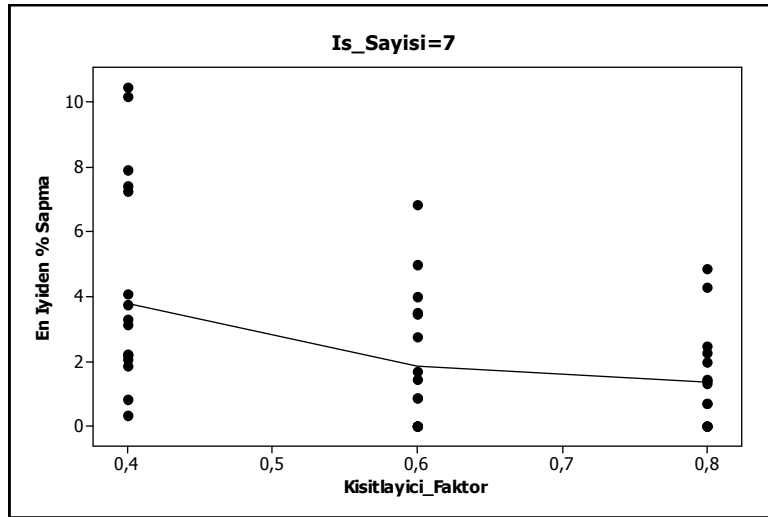
$F_{\alpha,(a-1)(b-1),ab(n-1)} = F_{0,05,6,168} \approx 2,10 > F_0 = 1,13$ olduğundan H_0 hipotezi kabul edilir; iş sayısı ve kısıtlayıcı faktör etkileşimi deney üzerinde etkili değildir.

İş sayısı 5, 7, 9 ve 11 olarak sabit tutulduğunda, sırasıyla Şekil 5.11, 5.12, 5.13 ve 5.14 teslim tarihini kısıtlandıran h faktörü ve en iyiden yüzde sapmalar arasındaki

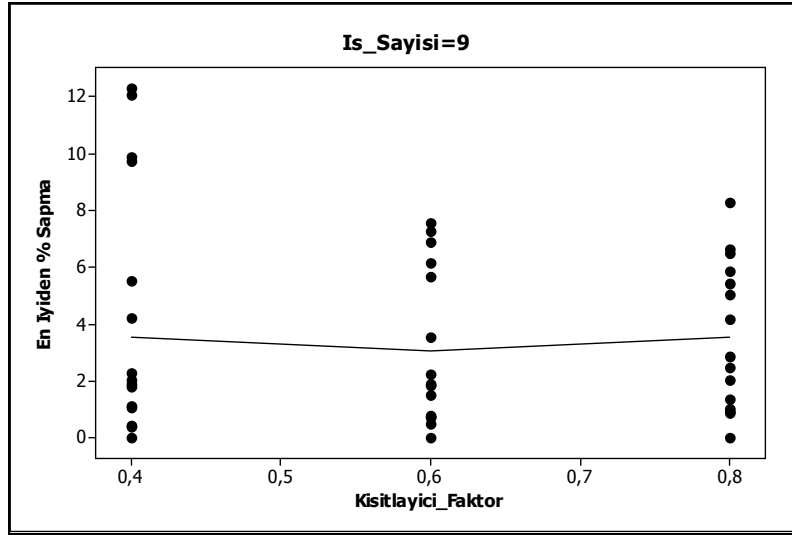
ilişkiyi göstermektedir.



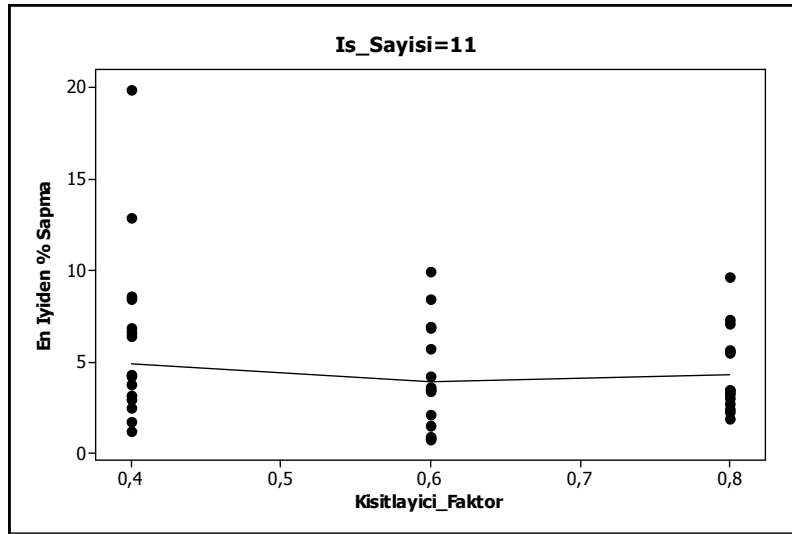
Şekil 5.11. TA'da $n=5$ için h ile en iyiden % sapma arasındaki ilişki



Şekil 5.12. TA'da $n=7$ için h ile en iyiden % sapma arasındaki ilişki

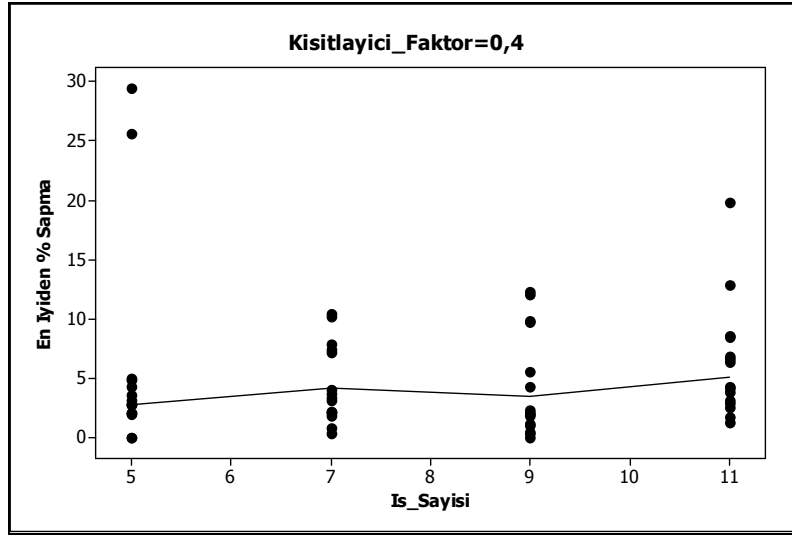


Şekil 5.13. TA'da $n=9$ için h ile en iyiden % sapma arasındaki ilişki

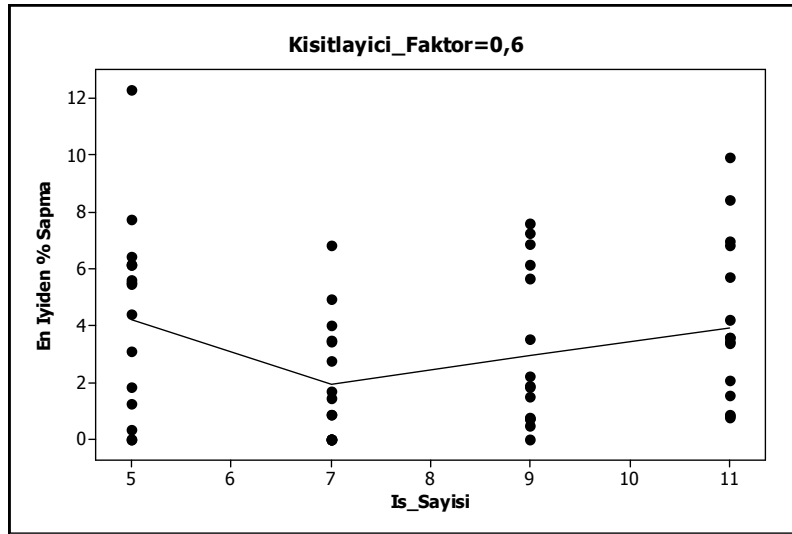


Şekil 5.14. TA'da $n=11$ için h ile en iyiden % sapma arasındaki ilişki

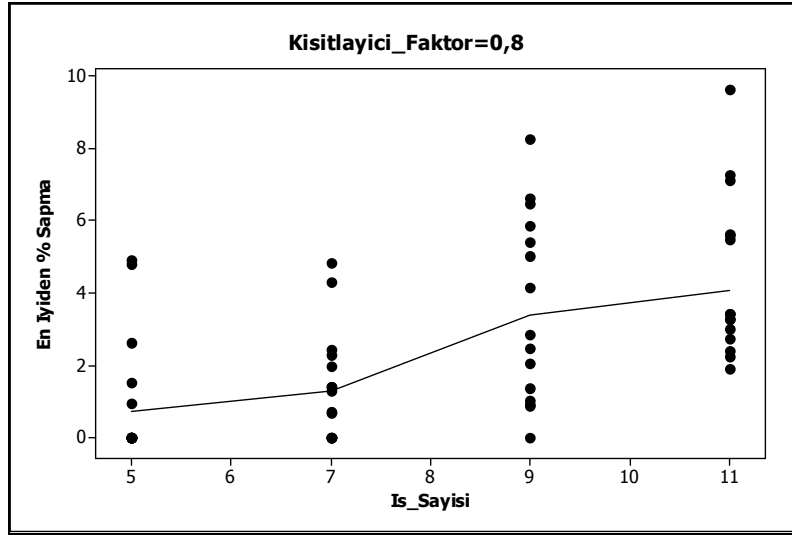
Genel olarak, iş sayısı sabit tutulduğunda teslim tarihinin dar olması en iyiden sapmayı arttırmaktadır. Benzer şekilde h faktörü sabit tutulduğunda iş sayısı ile en iyiden yüzde sapma arasındaki ilişkiyi gösteren grafikler Şekil 5.15 - Şekil 5.17'dir.



Şekil 5.15. TA'da $h=0,4$ için n ile en iyiden % sapma arasındaki ilişki

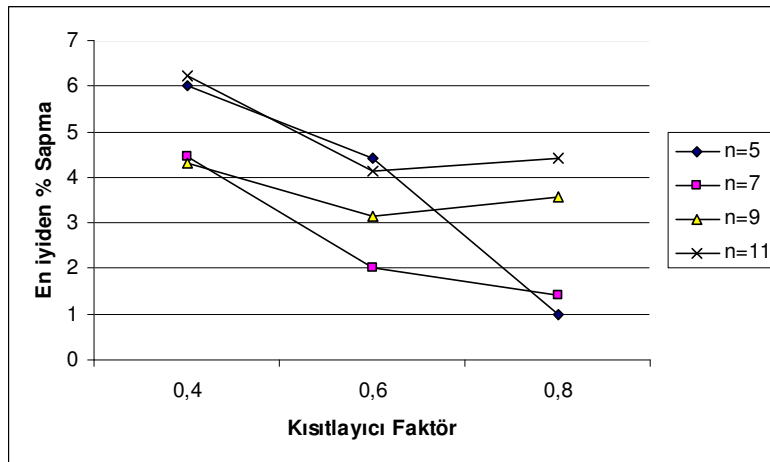


Şekil 5.16. TA'da $h=0,6$ için n ile en iyiden % sapma arasındaki ilişki



Şekil 5.17. TA’da $h=0,8$ için n ile en iyiden % sapma arasındaki ilişki

Şekil 5.18’de ise tek bir grafikte iş sayısı faktörünün tüm seviyeleri için kısıtlayıcı faktör ile en iyiden % sapma arasındaki ilişki görülmektedir.



Şekil 5.18. TA’da tüm iş sayıları için h ile en iyiden % sapma arasındaki ilişki

Tabu listesinin uzunluğu ve iterasyon sayısı değiştiğinde TA algoritmasının davranışını değerlendirmek için bir varyans analizi (ANOVA) daha yapılmıştır. Tasarlanan bu deneyde $n=11$ ve $h=0,8$ olarak alınarak yine 15 örnek seti

incelenmiştir. 2. deneyde kullanılan faktörler ve seviyeleri Çizelge 5.7’de görülmektedir.

Çizelge 5.7. TA algoritmasının 2. deneyindeki faktörler ve seviyeleri

Faktör	Kısaltma	Seviye
Tabu Listesinin Uzunluğu	Tabu_Listesi	5
		11
		21
		38
İterasyon Sayısı	İterasyon_Sayisi	10
		100
		1 000

Bu deneyde, her bir eşleşme durumu için 15 örnek çözülmüştür. Böylece toplam koşturma sayısı $15 \times 12 = 180$ olmuştur.

Çizelge 5.8 tüm modelin sonuçlarını göstermektedir. Aşağıda ise deney için kurulan üç hipotez yer almaktadır. Bu hipotezler sırasıyla tabu listesinin uzunluğunun, iterasyon sayısının ve bu iki faktörün etkileşiminin deney üzerinde etkili olup olmadığının sorgulanmasına yöneliktir.

$$H_0 : \tau_i = 0$$

$$H_1 : \tau_i \neq 0;$$

$$H_0 : \beta_j = 0$$

$$H_1 : \beta_j \neq 0;$$

$$H_0 : (\tau\beta)_{ij} = 0$$

$$H_1 : (\tau\beta)_{ij} \neq 0;$$

Çizelge 5.8. TA algoritmasının 2. deneyi için varyans analizi

Değişim Kaynağı	Kareler Toplamı	Serbestlik Derecesi	Ortalama Kare	F ₀	P Değeri
Tabu Listesi	3,075	3	1,025	0,13	0,945
İterasyon Sayısı	13,514	2	6,757	0,83	0,440
Etkileşim	2,303	6	0,384	0,05	1,000
Hata	1374,906	168	8,184		
Toplam	1393,798	179			

Tüm hipotezler %5 anlamlılık düzeyinde ($\alpha = 0,05$) değerlendirildiğinde aşağıdaki sonuçlara varılır:

$F_{\alpha,a-1,ab(n-1)} = F_{0,05,3,168} \approx 2,60 > F_0 = 0,13$ olduğundan H_0 hipotezi kabul edilir; tabu listesinin uzunluğu deney üzerinde etkili değildir.

$F_{\alpha,b-1,ab(n-1)} = F_{0,05,2,168} \approx 3,00 > F_0 = 0,83$ olduğundan H_0 hipotezi kabul edilir; iterasyon sayısı deney üzerinde etkili değildir.

$F_{\alpha,(a-1)(b-1),ab(n-1)} = F_{0,05,6,168} \approx 2,10 > F_0 = 0,05$ olduğundan H_0 hipotezi kabul edilir; tabu listesinin uzunluğu ve iterasyon sayısı etkileşimi deney üzerinde etkili değildir.

Bu deney sonucunda belirlenen seviyelerdeki tabu listesinin uzunluğunun ve iterasyon sayısının deney üzerinde etkili olmadığı ortaya çıkmıştır.

5.3. TA Algoritması ile Sağlanan İyileştirme

Önceki bölümlerde bahsedildiği gibi TA algoritması geliştirilen *SAPT* sezgiselinden elde edilen son çözümü başlangıç çözümü olarak alır ve bu çözümü iyileştirmeye çalışır. Bu kısımda TA ile sağlanan iyileştirme incelenmiş ve bu açıdan performansı ölçülmüştür. Çizelge 5.9'da daha önce hem *SAPT* sezgiselinin hem de TA algoritmasının deneylerinde kullanılan 180 örnek için bu iki yöntem arasındaki ilişki görülmektedir.

Çizelge 5.9. TA algoritması için deney sonuçları – 2

İş Sayısı	h faktörü	$SAPT$ sezgiseliyle en iyiye ulaşma sayısı	$SAPT$ sezgiselini iyileştirme sayısı	$SAPT$ sezgiselini ortalama iyileştirme (%)
5	0,4	1	5	1,96
	0,6	1	8	2,57
	0,8	10	2	1,38
7	0,4	0	10	2,68
	0,6	4	9	3,24
	0,8	3	9	2,82
9	0,4	0	12	4,66
	0,6	1	12	3,71
	0,8	1	10	2,73
11	0,4	0	15	5,82
	0,6	0	15	4,61
	0,8	0	14	4,52
Ortalama				3,39

Çizelge 5.9’da iş sayısının artışıyla birlikte TA’nın daha fazla örneği iyileştirdiği görülmektedir. 3. sütunda $SAPT$ sezgiseliyle bulunan en iyi çözümlerin sayısı yer almaktadır. Açıktır ki, en iyi çözümler iyileştirilemeyeceği için en düşük iyileştirme sayısı, $n=5$ ve $h=0,8$ iken karşımıza çıkar çünkü $SAPT$ sezgiseli zaten 10 örnekte en iyi çözümü bulmuştur, geriye kalan 5 örneğin ise ikisi iyileştirilmiştir. Genel olarak ortalama %3,39’luk iyileştirme sağlanmıştır. İyileştirilen 121 örneğin 19’unda en iyi çözüme ulaşılmıştır.

6. KULLANICIYA YÖNELİK ARAYÜZ OLUŞTURMA

6.1. Arayüz Hakkında Kısa Bilgi

Arayüz, kullanıcı ile etkileşime geçen, bir şeyler hazırlamasını, müzik dinlemesini, video izlemesini, veri girebilmesini vs. sağlayarak kullanıcıya yardımcı olan araçların bütünüdür. Kullanıcıya yönelik tasarlanan arayüz, görsel ya da şematik, basit ya da karmaşık olabilir.

6.2. Arayüzü Oluşturan Sayfalar

Bu tezde, kullanıcıya yönelik olarak geliştirilen arayüz toplam üç sayfadan oluşmaktadır. Bunlardan ilki kullanıcının yapmak istediği işlemlere geçişin sağlandığı açılış sayfası niteliğindedir. Diğer iki sayfa ise kullanıcının bu işlemlerin yapabilmesine olanak sağlar. Bu sayfalardan sunulan çeşitli örnekler, kullanımıyla ilgili daha ayrıntılı bilgi verecektir. Şekil 6.1’de ana sayfa görülmektedir.



Şekil 6.1. İş çizelgeleme programı–ana sayfa

Bu sayfada kullanıcıya üç seçenek sunulmaktadır. Bunlardan ilki, belirli kısıtlar altında önceden hazırlanmış olan örnek setlerini çalıştırabileceği ‘Örneklerden Çalıştırma Sayfası’ na geçiş; ikincisi, kendisinin gireceği değerlerle hesaplama yaptırabileceği ‘Girilen Değerlerle Çalıştırma Sayfası’ na geçiş; sonuncusu ise programdan çıkıştır. Şekil 6.2’de, yukarıda bahsedilen ayrıca bu çalışmada yazılan programların çalıştırılması için faydalanılan örnek setlerinin kullanıldığı ilk sayfanın açılış görüntüsü yer almaktadır:

Şekil 6.2. İş çizelgeleme programı-örneklerden çalıştırma sayfası (açılış ekranı)

Burada kullanıcının yapması gereken, geliştirilen yeni sezgiseli çalıştırmak için iş sayısı (n), örnek numarası (k) ve teslim tarihini kısıtlandırmak için kullanılan katsayı faktörünü (h); TA’yı çalıştırmak için de bunlara ilaveten tabu listesi uzunluğunu (l) ve maksimum iterasyon sayısını girmektir. Kullanıcı, bu verilerin girişlerinin yapıldığı panelin altındaki düğmelerden istediğine basarak *SAPT* sezgiselini ve/veya

TA'yı başlatabilir. Böylece istenilen numaralı örnek çalıştırılarak n sayıdaki iş, algoritmaya uygun şekilde kısıtlandırılmış bir teslim tarihine en yakın zamanda tamamlanır ve sonuçlar yan taraftaki panelde ekrana gelir. Sonuçların yer aldığı bu panelde bulunan amaç fonksiyon değeri, iş sırası, ilk işin başlama zamanı, programın çalışma süresi ve TA'nın *SAPT* sezgiseline göre yüzde kaç iyileşme sağladığı görülebilir. İş sayısı 10'dan fazla olduğunda metin kutusunun yanında 'kaydırıcı' elemanı ortaya çıkacaktır, böylece onarlı gruplar şeklinde iş sırası görülebilecektir. Ayrıca, yeni veriler girilmeden önce ekranın temizlenebilmesi için 'Temizle' düğmesine basılması yeterlidir. Bunların dışında, sayfada, sonuçlar tablosunun altında iki düğme daha yer almaktadır. Soldaki düğmeyle 'Girilen Değerlerle Çalıştırma Sayfası' na geçiş yapılabilir, sağdakiyle ise kullanıcı programdan çıkabilir. Şekil 6.3'te kullanıcı tarafından gerekli veriler girildiği takdirde ekrana gelen son görüntü yer almaktadır:

Şekil 6.3. İş çizelgeleme programı-örneklerden çalıştırma sayfası (çalıştırma sonrası)

‘Girilen Değerlerle Çalıştırma Sayfası’ na bu sayfadan geçilebileceği gibi ana sayfadan da geçilebilir. Şekil 6.4’te bu sayfanın açılış ekranı görülmektedir:

Şekil 6.4. İş çizelgeleme programı–girilen değerlerle çalıştırma sayfası (açılış ekranı)

Bazı farklılıklar dışında kullanıcının yapması gerekenler aynıdır. Şöyle ki, geliştirilen yeni sezgiseli çalıştırmak için iş sayısını (n) ve teslim tarihini (d); TA’yı çalıştırmak için de bunlara ilaveten tabu listesi uzunluğunu (l) ve maksimum iterasyon sayısını girmektir. Ancak bu sayfa, kullanıcının kendi verileriyle programı çalıştırabilmesine olanak vermesi için tasarlandığı için işlerin hazırlık ve işlem sürelerinin veri olarak girilmesi lazımdır. Özellikle iş sayısı fazla olduğunda girişinin kolay olması açısından bu verilerin açılacak bir Excel sayfasında, hazırlık sürelerinin matris olarak ve işlem sürelerinin tek satır halinde ayrı ayrı girilebileceği iki alt sayfaya kaydedilmesi gerekmektedir. Excel’de veriler girildikten sonra bu sayfanın yolu panelde yer alan metin kutusuna yazılmalıdır. Sonrasında ise kullanıcı yine panelin altındaki düğmelerden istediğine basarak SAPT sezgiselini ve/veya TA’yı başlatabilir.

Böylece istenilen numaralı örnek çalıştırılarak n sayıdaki iş, algoritmaya uygun şekilde belirlenen teslim tarihine en yakın zamanda tamamlanır ve sonuçlar yan taraftaki panelde ekrana gelir. Sonuçların yer aldığı bu panelde bulunan amaç fonksiyon değeri, iş sırası, ilk işin başlama zamanı, programın çalışma süresi ve TA'nın *SAPT* sezgiseline göre yüzde kaç iyileşme sağladığı görülebilir. 'Örneklerden Çalıştırma Sayfası' nda olduğu gibi iş sayısı 10'dan fazla olduğunda metin kutusunun yanında 'kaydırıcı' elemanı ortaya çıkacaktır ve ekranın temizlenebilmesi için 'Temizle' düğmesine basılmalıdır. Bunların dışında, sayfada, sonuçlar tablosunun altında iki düğme daha yer almaktadır. Soldaki düğmeyle 'Örneklerden Çalıştırma Sayfası' na geçiş yapılabilir, sağdakiyle ise kullanıcı programdan çıkabilir. Şekil 6.5'te kullanıcı tarafından gerekli veriler girildiği takdirde ekrana gelen son görüntü yer almaktadır:

Şekil 6.5. İş çizelgeleme programı–girilen değerlerle çalıştırma sayfası (çalıştırma sonrası)

6.3. Arayüzde Kullanılan Uyarı/Hata Mesajları ve İpucu Metinleri

Tasarlanan arayüzde kullanıcıya ipucu veren metinler ile onu uyarma, hatasını belirtme amacı taşıyan çeşitli mesajlar hazırlanmıştır. İpucu metinleri bazı düğmelere farenin yaklaştırılmasıyla birlikte ekranda belirir. Bununla birlikte, hata mesajları çözümü mümkün olmayan değerler girildiğinde ortaya çıkar. Uyarı mesajı ise sadece kullanıcı programdan çıkış yapmak için düğmeye bastığında ona geri dönüş imkânı sağlamak için hazırlanmıştır. Aşağıda, programda yer aldığı biçimiyle bu metinler ve mesajlar yer almaktadır.

İpucu Metinleri:

- Deger girerek çalistirmek için basiniz.
- Mevcut örnek setlerini çalistirmek için basiniz.
- Örnek setlerini çalistirma sayfasina geçis
- Girilen degerlerle çalistirma sayfasina geçis

Uyarı Mesajı:

- Programdan çıkmak istediginize emin misiniz?

Hata Mesajları:

- İlk iki girdi için tamsayı degerler girmelisiniz!
- Sifir degeri girilemez!
- Is sayisi için 3 ten büyük bir deger giriniz!
- Sayisal bir deger girmelisiniz!
- Tamsayı degerler girmelisiniz!
- Is sayisini yanlis girdiniz!

7. SONUÇ VE ÖNERİLER

Bu bölümde, önceki bölümlerde yapılan çalışmalarla elde edilen sonuçlara dayalı olarak değerlendirmeler yapılmıştır. Bu çalışmada ele alınan problemi diğer E/G problemlerinden ayıran nokta hem sıraya-bağımlı hazırlık sürelerini ihtiva etmesi hem de kısıtlandırılmış teslim tarihli olmasıdır. Bu farklılıklarla birlikte problemin çözümü, hazırlık sürelerinin işlem süreleri içinde yer aldığı ve/veya kısıtlandırılmamış teslim tarihli problemlerinkinden çok daha zor hale gelmiştir. Özetle, bu çalışmada kısıtlandırılmış ortak teslim tarihli, eşit E/G cezaları ve sıraya-bağımlı hazırlık süreleri içeren tek makineli E/G problemi ele alınmıştır.

Bu bölüm üç kısımdan oluşmaktadır. İlk kısımda, çalışmanın katkıları anlatılmıştır. İkinci kısımda, geliştirilen KTP, *SAPT* sezgiseli ve TA algoritmalarının performansları sunulmuştur. Üçüncü kısımda da daha sonraki çalışmalarda nelerin ele alınabileceğine ilişkin öneriler yapılmıştır.

7.1. Çalışmanın Katkıları

Bu çalışmanın temel katkıları aşağıda verilmiştir:

- 1) Sıraya-bağımlı hazırlık süreleri içeren kısıtlandırılmış ortak teslim tarihli tek makineli E/G probleminin küçük boyutluları için en iyi çözümleri makul süre içinde bulan bir karışık tamsayılı programlama (KTP) geliştirilmiştir.
- 2) Problem için oldukça iyi sonuçlar bulan bir sezgisel (*SAPT*) geliştirmiştir.
- 3) *SAPT* sezgiseliyle elde edilen en iyi çözümü başlangıç çözümü olarak alan ve bunu iyileştirerek en iyi ya da en iyiye yakın çözümler bulan bir TA algoritması geliştirilmiştir.
- 4) Kullanıcının programı hem daha rahat hem de zevkle kullanabilmesi için bir arayüz tasarımı yapılmıştır.

7.2. Sonular

Önceki bölümlerde elde edilen sonuçlara dayalı olarak aşağıdaki sonuçlar çıkarılabilir:

- 1) Bu alışmada geliştirilen KTP ile 11 işe kadar olan büyüklükteki E/G problemlerinin makul işlem sürelerinde en iyi sonuçları bulunmuştur.
- 2) Geliştirilen *SAPT* sezgiseli, alışmada ele alınan E/G problemi için iyi sonuçlar bulmuştur. 11 işe kadarki problemler için *SAPT* ile bulunan sonuçların en iyi sonuçlardan ortalama sapması yaklaşık %5,5'tir.
- 3) İş sayısı *SAPT* sezgiseli üzerinde önemli bir etkidir. 5, 7 ve 9 işli problemler için ortalama sapma birbirine yakınken 11 işli problem için sapma artış göstermiştir.
- 4) Teslim tarihini kısıtlandırmak için kullanılan h faktörü de *SAPT* sezgiseli üzerindeki diğer önemli etkidir. En iyi özümde sapma birkaç istisna dışında daima en küçük h faktörüyle yapılan deneylerde (yani ortak teslim tarihinin en küçük olduğu durumda) ortaya çıkmıştır.
- 5) 11 ve daha az işli problemler için TA sonuçlarının en iyi sonuçlardan ortalama sapması yaklaşık %2'dir. Açıktır ki, TA algoritması *SAPT* sezgiselini geliştirmiş, en iyiye oldukça yakın özümler bulmuştur.
- 6) TA üzerinde iş sayısı ve h faktörünün etkili olduğu ancak tabu listesinin uzunluğunun ve iterasyon sayısının etkili olmadığı deney tasarımı sonucunda bulunmuştur.
- 7) TA algoritmasının 180 işe kadar olan problemleri makul süreler içinde özebildiği ancak daha büyük boyutlu problemlerde ok fazla zamana ihtiyaç duyduğu gözlenmiştir.

7.3. Gelecek Çalışmalar için Öneriler

Bu çalışmada, kısıtlandırılmış ortak teslim tarihli ve sıraya-bağımlı hazırlık süreleri içeren E/G problemi ele alınmıştır. Genel olarak, sıraya-bağımlı hazırlık sürelerini konu alan fazla çalışma olmadığından, gelecek çalışmalarda dikkate alınmak üzere aşağıda önerilen konular, sıraya-bağımlı hazırlık süreleri içeren problemlerden oluşmaktadır.

1) *Modelde eşit olmayan cezaların ele alınması*: Kullanılan cezaların tipi problemin karmaşıklığını büyük ölçüde etkiler. İşe-bağımlı cezalar içeren problemler diğer ceza tiplerini içerenlerden çok daha zordur. Bu durumda, *SAPT* sezgiseli söz konusu olan yeni ceza tiplerine göre düzenlenmelidir.

2) *Farklı teslim tarihlerinin ele alınması*: Bu çalışmada, farklı teslim tarihlerinin özel bir durumu olan ortak teslim tarihli problem incelenmiştir. Genel anlamdaki farklı teslim tarihleri araştırılabilir. Bu durumda boş zamanların yerleştirilmesi gerekeceği için probleme yeni bir zorluk eklenmiş olur.

3) *Paralel makineli sistemler incelenmesi*: Bu problem çerçevesinde, özdeş paralel makineler, eşit cezalar, kısıtlandırılmamış ortak teslim tarihi ve sıraya-bağımlı hazırlık süreleri ele alınabilir.

KAYNAKLAR

1. Pinedo, M., "Scheduling: Theory, Algorithms, and Systems", 2nd ed., **Prentice Hall**, New Jersey, 14-16, 79, 80 (2002).
2. Li, C. L., Cheng, T. C. E. and Chen, Z. L., "Single-machine scheduling to minimize the weighted number of early and tardy agreeable jobs", **Computers and Operations Research**, 22: 205-219 (1995).
3. Rabadi G., "Minimizing the total earliness and tardiness for a single machine scheduling problem with a common due date and sequence dependent setup times", Doktora Tezi, **University of Central Florida**, USA, 35-37, 43, 50-53, 63-87 (1999).
4. Baker, K. R. and Scudder, G. D., "Sequencing with earliness and tardiness penalties: A review", **Operations Research**, 38 (1): 22-36 (1990).
5. Bagchi, U., Sullivan R. S. and Chang, Y. L., "Minimizing mean absolute deviation of completion times about a common due date", **Naval Research Logistics**, 33: 227-240 (1986).
6. Bagchi, U., Chang, Y. L. and Sullivan R. S., "Minimizing absolute and squared deviations of completion times with different earliness and tardiness penalties and a common due date", **Naval Research Logistics**, 34: 739-751 (1987).
7. Raghavachari, M., "A V-shape property of optimal schedule of jobs about a common due date", **European Journal of Operational Research**, 23: 401-402 (1986).
8. Hall, N. G., Kubiak, W. and Sethi, S. P., "Deviation of completion times about a restrictive common due date", **Working Paper**, College of Business, The Ohio State University, Columbus, 89-19 (1989).
9. Kanet, J. J., "Minimizing the average deviation of job completion times about a common date", **Naval Research Logistics Quarterly**, 28: 643-651 (1981).
10. Szwarc, W., "Single machine scheduling to minimize absolute deviation of completion times from a common due date", **Naval Research Logistics**, 36: 663-673 (1989).
11. Sundararaghavan, P. S. and Ahmed, M. U., "Minimizing the sum of absolute lateness in single machine and multimachine scheduling", **Naval Research Logistics Quarterly**, 31: 325-333 (1984).
12. Baker, K. R. and Chadowitz, A., "Algorithms for minimizing earliness and tardiness penalties with a common due date", **Working Paper**, Amos Tuck School of Business Administration, Dartmouth College, Hanover, N. H., 240 (1989).

13. Rabadi, G., Mollaghasemi, M. and Anagnostopoulos, G. C., "A branch-and-bound algorithm for the early/tardy machine scheduling problem with a common due-date and sequence-dependent setup time", *Computers and Operations Research*, 31: 1727-1751 (2004).
14. Emmons, H., "Scheduling to a common due date on parallel uniform processors", *Naval Research Logistics*, 34: 803-810 (1987).
15. Panwalkar, S. S., Smith, M. L. and Seidmann, A., "Common due date assignment to minimize total penalty for the one machine scheduling problem", *Operations Research*, 30: 391-399 (1982).
16. Chang, S. S. and Joo, U. G., "A single machine scheduling problem with earliness/tardiness and starting time penalties under a common due date", *Computers & Operations Research*, 19 (8): 753-766 (1992).
17. Panwalkar, S. S., Rajagopalan, R., "Single-machine sequencing with controllable processing times", *European Journal of Operational Research*, 59: 298-302 (1992).
18. Liman, S. D., Ramaswamy, S., "Earliness-tardiness scheduling problems with a common delivery window", *Operations Research Letters*, 15: 195-203 (1994).
19. Liman, S. D., Panwalkar, S. S. and Thongmee, S., "Determination of common due window location in a single machine scheduling problem", *European Journal of Operational Research*, 93: 68-74 (1996).
20. Ventura, J. A., Kim, D., Garriga, F., "Single machine earliness-tardiness scheduling with resource-dependent release dates", *European Journal of Operational Research*, 142: 52-69 (2002).
21. Bagchi, U., Sullivan R. S., Chang, Y.-L., "Minimizing mean squared deviation of completion times about a common due date", *Management Science*, 33: 894-906 (1987).
22. Eilon, S. and Chowdhury, I., "Minimizing waiting time variance in the single machine problem", *Management Science*, 23: 567-75 (1977).
23. Kanet, J. J., "Minimizing variation of flow time in single machine systems", *Management Science*, 27: 1453-59. (1981).
24. Vani, V. and Raghavachari, M., "Deterministic and random single machine scheduling with variance minimization", *Operations Research*, 35: 111-20 (1987).
25. De, P., Ghosh, J. B. And Wells, C. E., "A note on the minimization of mean squared deviation of completion times about a common due date", *Management*

- Science*, 35:1143-1147 (1989a).
26. Leung J.Y.-T., "A dual criteria sequencing problem with earliness and tardiness penalties", *Naval Research Logistics*, 49: 422-431 (2002).
 27. Cheng, T., "An algorithm for the CON due date determination and sequencing problem", *Computers & Operations Research*, 14: 537-542 (1987).
 28. Quaddus, M., "A generalized model of optimal due-date assignment by linear programming", *Journal of the Operational Research Society*, 38: 353-359 (1987).
 29. Bector, C., Gupta, Y. and Gupta, M., "Determination of an optimal common due date and optimal sequence in a single machine job shop", *International Journal of Production Research*, 26: 613-28 (1988).
 30. Hall, N. G., Posner, M. E., "Weighted deviation of completion times about a common due date", *Working Paper*, College of Business, The Ohio State University, *Columbus*, 89-15 (1989).
 31. Hino, C. S., Ronconi, D. P., Mendes, A. B., "Minimizing earliness and tardiness penalties in a single-machine problem with a common due date", *European Journal of Operational Research*, 160: 190-201 (2005).
 32. Biskup, D., Feldmann, M., "Benchmarks for scheduling on single/machine against restrictive and unrestrictive common due date", *Computers & Research*, 28: 787-801 (2001).
 33. Feldmann, M., Biskup, D., "Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches", *Computers & Industrial Engineering*, 44: 307-323 (2003).
 34. Hall, N. G., "Single and multi-processor models for minimizing completion time variance", *Naval Research Logistics Quarterly*, 33, 49-54 (1986).
 35. Cheng, T. C. E., "A heuristic for common due date assignment and job scheduling on Parallel Machines", *Journal of Operational Research Society*, 40: 1129-1135 (1989).
 36. Adamopoulos, G. I. and Pappis, C. S., "Scheduling under a common due date on parallel unrelated machines", *European Journal of Operational Research*, 105: , 495-501 (1998).
 37. Biskup, D., Cheng T. C. E., "Multiple-machine scheduling with earliness, tardiness and completion time penalties", *Computers & Research*, 26: 45-57 (1999).
 38. Panwalkar, S. S., Liman, S. D., "Single operation earliness-tardiness scheduling

- with machine activation cost”, *IIE Transactions*, 34: 509-513 (2002).
39. Cheng, T., “Optimal common due date with limited completion time deviation”, *Computers & Operations Research*, 15: 91-96 (1988).
 40. Seidmann, A., Panwalkar, S. S. and Smith, M. L., “Optimal assignment of due-dates for a single processor scheduling problem”, *International Journal of Production Research*, 19 (4): 393-399 (1981).
 41. Garey, M. R., Tarjan, R. E. and Wilfong, G. T., “One processor scheduling with symmetric earliness and tardiness penalties”, *Mathematics of Operations Research*, 13: 330-348 (1988).
 42. Yano, C. A. and Kim, Y. D., “Algorithms for single machine scheduling problems minimizing tardiness and earliness”, *Technical Report*, Department of Industrial Engineering, University of Michigan, *Ann Arbor*, 86 (1986).
 43. Abdul-Razaq, T. and Potts, C., “Dynamic programming state-space relaxation for single-machine scheduling”, *Journal of Operational Research Society*, 39: 141-152 (1988).
 44. Ow, P. S. and Morton, T. E., “Filtered beam search in scheduling”, *International Journal of Production Research*, 26: 35-62 (1988).
 45. Ow, P. S. and Morton, T. E., “The single machine early/tardy problem”, *Management Science*, 35: 177-191 (1989).
 46. Fry, T.D., Armstrong. R.D., Blackstone, J.H., “Minimizing weighted absolute deviation in single machine scheduling”, *IIE Transactions*, 19: 445-450 (1987).
 47. Gupta, S. and Sen, T., “Minimizing a quadratic function of job lateness on a single machine”, *Engineering Costs Production Economics*, 7: 181-194 (1983).
 48. Valente, J.M.S., Alves, R.A.F.S., “Filtered and recovering beam search algorithms for early/tardy scheduling problem with no idle time”, *Computers and Industrial Engineering*, 48: 363-375 (2005).
 49. Yano, C. A. and Kim, Y. D., “Algorithms for a class of single-machine weighted tardiness and earliness problems”, *European Journal of Operational Research*, 52: 167-178 (1991).
 50. Szwarc, W. and Mukhopadhyay, S. K., “Optimal timing schedules in earliness-tardiness single-machine sequencing”, *Naval Research Logistics*, 42: 1109-1114 (1995).
 51. Keyser, T. K. and Sarper, H., “A heuristic solution of the E/G problem with waiting costs and non-zero release times”, *Computers & Industrial Engineering*, 21: 297-301 (1991).

52. Soroush, S. M. and Fredendall, L. D., "The stochastic single machine scheduling problem with earliness-tardiness costs", *European Journal of Operational Research*, 77: 287-302 (1994).
53. Koulamas, C., "Single machine scheduling with time windows and earliness/tardiness penalties", *European Journal of Operational Research*, 91: 190-202 (1996).
54. Ahmadi, R. and Bagchi U., "Single machine scheduling to minimize earliness subject to deadlines", *Working Paper*, Department of Management, University of Texas, *Austin*, 86/86-4-17 (1986a).
55. Ahmadi, R. and Bagchi U., "Just-in-time scheduling in single machine systems", *Working Paper*, Department of Management, University of Texas, *Austin*, 85/86-4-21 (1986b).
56. Chand, S. and Schneeberger, H., "Single machine scheduling to minimize weighted earliness subject to no tardy jobs", *European Journal of Operational Research*, 34: 221-230 (1988).
57. Bagchi, U., "Due date or deadline assignment to multi-job orders to minimize total penalty in the one machine scheduling problem", Presented at the *ORSA/TIMS Joint National Conference*, St. Louis, 210-218 (1987).
58. Baker, K. R., "Introduction to sequencing and scheduling", *John Wiley and Sons*, Inc., New York, 94-96 (1974).
59. French, S., "Sequencing and scheduling: An introduction to the mathematics of the job-shop", *John Wiley and Sons*, New York, 199 (1982).
60. Allahverdi, A., Gupta, J. N. D., Aldowaisan, T., "A review of scheduling research involving setup consideration", *OMEGA*, 27: 219-239 (1999).
61. Coleman, B. J., "A simple model for optimizing the single machine early/tardy problem with sequence-dependent setups", *Production and Operation Management*, 1: 225-228 (1992).
62. Chen, Z. L., "Scheduling with batch setup times and earliness-tardiness penalties", *European Journal of Operational Research*, 96: 518-537 (1997).
63. Hall, N. G. and Posner, M. E., "Earliness-tardiness scheduling problems, I: Weighted deviation of completion times about a common due date", *Operations Research*, 39 (5): 836-846 (1991).
64. Hall, N. G., Kubiak, W. and Sethi, S. P., "Earliness-tardiness scheduling problems, I: Weighted deviation of completion times about a restrictive common due date", *Operations Research*, 39 (5): 847-856 (1991).

65. Liman, S. D. and Lee C-Y., "Error bound of a heuristic for the common due date scheduling problem", *ORSA Journal on Computing*, 5 (4): 420-425 (1993).
66. James, R. J. W., "Using tabu search to solve the common due date early/tardy machine scheduling problem", *Computers & Operations Research*, 24 (3): 199-208 (1997).
67. Chen, Z. L. and Powell W. B., "A column generation based decomposition algorithm for a parallel machine just-in-time scheduling problem", To appear, *European Journal of Operational Research* (1997).
68. Liaw, C-F., "A branch-and-bound algorithm for the single-machine earliness and tardiness scheduling problem", *Computers & Operations Research*, 26: 679-693 (1999).
69. Morton, T. E. and Pentico, D. W., "Heuristic scheduling systems with applications to production systems and project management", *John Wiley and Sons, Inc.*, New York, 172-181 (1993).
70. Balakrishnan, N., Kanet, J. J. and Sridharan, S. V., "Early/tardy scheduling with sequence dependent setup on uniform parallel machines", *Computers & Operations Research*, 26: 127-141 (1999).
71. Glover, F., "Future paths for integer programming and links to artificial intelligence", *Computers and Operations Research*, 13: 533-549 (1986).
72. Hertz, A., and De, W. D., "The tabu search metaheuristic: How we used it", *Annals of Mathematics and Artificial Intelligence*, 1: 111-121 (1991).
73. Glover, F., "Tabu search - Part I", *ORSA Journal of Computing*, 1 (3): 190-206 (1989).
74. Barnes, J.W., Laguna, M., "A review and synthesis of tabu search applications to production scheduling problems", *Technical Report*, University of Texas at Austin, *ORP* 91-05 (1991).
75. Widmer, M., Hertz, A., "A new heuristic method for the flow shop sequencing problem", *European Journal of Operational Research*, 41: 186-193 (1989).
76. Kim, Y.-D., "Heuristics for flow shop scheduling problems minimizing mean tardiness", *Journal of the Operational Research Society*, 44: 19-28 (1993).
77. Taillard, E., "Some efficient heuristic methods for the flow shop sequencing problem", *European Journal of Operational Research*, 41: 65-74 (1990).
78. Reeves, C. R., "Improving the efficiency of tabu search for machine sequencing problems", *Journal of the Operational Research Society*, 44: 375-382 (1993).

79. Adenso-Diaz, B., "Restricted neighbourhood in the tabu search for the flowshop problem", *European Journal of Operational Research*, 62: 27-37 (1992).
80. Widmer, M., "Job shop scheduling with tooling constraints: A tabu search approach", *Journal of the Operational Research Society*, 42: 75-82 (1991).
81. Barnes, J. W., Chambers, J. B., "Solving the job shop scheduling problem with tabu search", *IIE Transactions*, 27: 257-263 (1995).
82. Sun, D., Batta, R., Lin, L., "Effective job shop scheduling through active chain manipulation", *Computers and Operations Research*, 22: 159-172 (1995).
83. Barnes, J. W., Laguna, M., "Solving the multiple-machine weighted flow time problem using tabu search", *IIE Transactions*, 25: 121-128 (1993).
84. Hübscher, R., Glover, F., "Applying tabu search with influential diversification to multiprocessor scheduling", *Computers and Operations Research*, 21: 877-884 (1994).
85. Laguna, M., Gonzalez, V. J. L., "A search heuristic for just-in-time scheduling in parallel machines", *Journal of Intelligent Manufacturing*, 2: 253-260 (1991).
86. Laguna, M., Barnes, J. W., Glover, F., "Tabu search methods for a single machine scheduling problem", *Journal of Intelligent Manufacturing*, 2: 63-74 (1991).
87. Laguna, M., Glover, F., "Integrated target analysis and tabu search for improved scheduling systems", *Expert Systems with Applications*, 6: 287-297 (1993).
88. James, R. J. W., Buchanan, J.T., "Performance enhancements to tabu search for the early-tardy scheduling problem", *European Journal of Operational Research*, 106: 254-265 (1998).
89. James, R. J. W., Buchanan, J.T., "A neighbourhood scheme with a compressed solution space for the early/tardy scheduling problem", *European Journal of Operational Research*, 102: 513-527 (1997).
90. Reeves, C. R., "Improving the efficiency of tabu search for machine sequencing problems", *Journal of the Operational Research Society*, 44: 375-382 (1993).
91. Glover, F., Laguna, M., "Tabu Search", *Modern Heuristic Techniques for Combinatorial Problems*, ed. C. R. Reeves. Blackwell Scientific Publications, 70-150 (1993).
92. Barnes, J. W., Laguna, M., "A tabu search experience in production scheduling", *Annals of Operations Research*, 41: 141-156 (1993).

93. Hao, Q., Yang, Z., Wang, D. and Li, Z., "Common due-date determination and sequencing using tabu search", *Computers and Operations Research*, 23: 409-417 (1996).
94. Glover, F., "Tabu Search - Part II", *ORSA Journal on Computing*, 2 (1): 4-32 (1990).

EKLER

EK-1. AMPL diliyle yazılan model

```

set ISLER;

param Islem {ISLER};           #Islem süreleri
param N;                       #Islerin sayisi
param TeslimTarihi;            #Teslim tarihi
param Hazirlik {ISLER, ISLER}; #Hazirlik süreleri
param M :=1000;
param _total_solve_time ;      #Islem süresi (CPU)

var TamZamani {ISLER} >= 0;    #Tamamlanma zamani degiskeni
var Erken {ISLER} >= 0;        #Erkenlik degiskeni
var Gec {ISLER} >= 0;          #Geclik degiskeni
var Y {ISLER, ISLER} binary;  #Yij ikili degiskeni

minimize EG_cost: sum {i in ISLER}(Erken[i]+Gec[i]);

subject to constr1 {i in ISLER}: TamZamani[i] - Islem[i] >= 0;

subject to constr2 {i in ISLER}: TamZamani[i] - Gec[i] + Erken[i] - Teslimtarihi =0;

subject to constr3 {i in 1..N, j in i+1..N}:

    TamZamani[j] - TamZamani[i] + M*(1-Y[i,j]) - Islem[j] - Hazirlik[i,j] >= 0;

subject to constr4 {i in 1..N, j in i+1..N}:

    TamZamani[i] - TamZamani[j] + M*Y[i,j] - Islem[i] - Hazirlik[j,i] >= 0;

```

EK-2. KTP ile bulunan en iyi çözümler ($n=5$)

Problem No	h	En iyi Değer	En iyi Sıra	Simpleks İtr. Sayısı	D&S Düg. Sayısı	CPU (sn)
1	0,4	356	1-3-4-2-5	380	79	0,03125
	0,6	285	5-1-4-3-2	569	111	0,0625
	0,8	263	5-1-4-3-2	500	84	0,09375
2	0,4	543	5-4-3-2-1	565	140	0,078125
	0,6	478	1-4-3-5-2	751	155	0,09375
	0,8	445	2-1-3-5-4	761	141	0,078125
3	0,4	261	2-5-1-4-3	336	64	0,0625
	0,6	250	4-1-2-5-3	439	91	0,0625
	0,8	204	4-2-5-1-3	359	44	0,0625
4	0,4	345	5-4-2-1-3	265	61	0,078125
	0,6	282	1-5-4-2-3	330	61	0,046875
	0,8	267	3-1-4-2-5	463	56	0,0625
5	0,4	328	5-3-2-4-1	499	108	0,0625
	0,6	286	1-5-3-2-4	471	101	0,0625
	0,8	254	1-5-3-2-4	514	93	0,046875
6	0,4	361	4-2-5-3-1	435	81	0,078125
	0,6	328	3-4-2-5-1	493	83	0,0625
	0,8	317	3-4-2-5-1	582	97	0,0625
7	0,4	515	5-2-3-1-4	518	127	0,078125
	0,6	448	5-2-3-1-4	671	129	0,078125
	0,8	416	4-5-2-1-3	695	116	0,09375
8	0,4	522	2-5-1-4-3	615	174	0,078125
	0,6	420	3-2-5-1-4	756	164	0,09375
	0,8	393	3-2-5-1-4	716	127	0,0625
9	0,4	395	3-5-4-2-1	314	63	0,046875
	0,6	288	3-2-5-4-1	280	51	0,0625
	0,8	269	3-1-5-4-2	278	42	0,0625
10	0,4	391	3-5-1-2-4	508	116	0,0625
	0,6	355	2-5-1-3-4	679	120	0,078125
	0,8	330	2-4-1-3-5	617	100	0,0625
11	0,4	435	5-4-3-1-2	522	117	0,0625
	0,6	385	1-5-4-3-2	671	153	0,078125
	0,8	344	2-5-4-3-1	641	119	0,0625
12	0,4	550	2-4-3-1-5	453	99	0,0625
	0,6	456	2-5-4-3-1	763	132	0,0625
	0,8	420	5-2-4-3-1	782	126	0,078125
13	0,4	465	3-1-4-5-2	524	95	0,0625
	0,6	388	3-5-1-4-2	634	142	0,0625
	0,8	370	2-3-1-4-5	589	98	0,0625
14	0,4	272	2-4-5-3-1	519	101	0,0625
	0,6	228	4-5-3-2-1	585	138	0,0625
	0,8	209	1-4-5-3-2	547	97	0,0625
15	0,4	502	5-2-3-1-4	475	121	0,0625
	0,6	419	5-2-3-1-4	643	128	0,0625
	0,8	384	5-4-3-2-1	520	89	0,0625
Ortalama				538,38	105,2	0,067

EK-2. (Devam) KTP ile bulunan en iyi çözümler ($n=7$)

Problem No	h	En iyi Değer	En iyi Sıra	Simpleks İtr. Sayısı	D&S Düş. Sayısı	CPU (sn)
1	0,4	493	3-2-1-5-4-6-7	4732	995	0,359375
	0,6	401	7-3-2-1-5-4-6	6056	823	0,40625
	0,8	387	7-3-2-1-5-4-6	7149	993	0,46875
2	0,4	767	7-4-2-5-1-6-3	10432	2240	0,6875
	0,6	602	5-7-4-2-1-6-3	8150	1050	0,484375
	0,8	584	5-7-6-4-2-1-3	7476	810	0,421875
3	0,4	518	1-7-3-2-6-5-4	8377	1747	0,546875
	0,6	435	5-1-7-3-2-6-4	11760	1937	0,703125
	0,8	424	4-1-7-3-2-6-5	10899	1668	0,6875
4	0,4	722	7-3-4-5-2-1-6	7259	1367	0,5
	0,6	577	6-7-4-5-3-2-1	9125	1422	0,515625
	0,8	557	6-7-3-4-5-2-1	11464	1797	0,703125
5	0,4	900	3-6-1-2-4-5-7	11953	2766	0,796875
	0,6	777	7-3-6-1-2-5-4	14540	2560	0,921875
	0,8	763	7-4-5-1-2-3-6	15754	2701	0,953125
6	0,4	792	7-4-1-5-2-6-3	18101	5326	1,26562
	0,6	697	4-6-1-5-2-7-3	22909	5089	1,625
	0,8	682	3-7-6-1-5-2-4	21364	4877	1,4375
7	0,4	745	2-6-4-5-7-1-3	10625	2128	0,640625
	0,6	612	3-1-4-5-6-7-2	11332	1771	0,703125
	0,8	604	3-1-4-5-6-7-2	11444	1596	0,703125
8	0,4	589	2-1-7-6-3-5-4	5426	1144	0,375
	0,6	535	2-6-1-7-3-5-4	9174	1501	0,546875
	0,8	485	4-2-1-7-6-3-5	8181	1014	0,515625
9	0,4	828	1-5-4-3-2-7-6	15324	3986	1,03125
	0,6	718	6-1-5-4-3-7-2	16568	3615	1,1875
	0,8	684	6-1-5-4-3-7-2	19187	3465	1,17188
10	0,4	1039	2-3-4-7-1-6-5	14516	3552	1
	0,6	858	5-2-3-4-7-1-6	15244	2488	0,90625
	0,8	858	5-2-3-4-7-1-6	14781	2477	0,921875
11	0,4	893	3-4-1-7-6-5-2	12773	3151	0,90625
	0,6	743	3-4-7-1-6-5-2	15925	2992	1,01562
	0,8	722	2-3-4-1-7-6-5	16215	2819	1,01562
12	0,4	542	5-3-4-7-2-6-1	9664	2187	0,625
	0,6	472	1-5-3-4-2-6-7	12172	2299	0,75
	0,8	441	1-6-5-3-4-2-7	12038	1985	0,75
13	0,4	831	7-6-3-2-1-5-4	14575	3256	1,01562
	0,6	674	5-7-6-3-2-1-4	17759	3004	1,1875
	0,8	659	4-5-7-6-3-2-1	12466	2011	0,78125
14	0,4	671	7-2-3-1-5-4-6	9472	1924	0,625
	0,6	616	5-4-1-2-7-3-6	13935	2502	0,875
	0,8	572	6-3-2-7-1-5-4	11815	1660	0,671875
15	0,4	679	3-5-2-4-1-7-6	9745	2231	0,65625
	0,6	586	3-5-2-1-4-7-6	9149	4129	0,5625
	0,8	566	6-3-5-2-4-1-7	14093	2344	0,890625
Ortalama				12246,62	2386,64	0,789

EK-2. (Devam) KTP ile bulunan en iyi çözümler ($n=9$)

Problem No	h	En iyi Değer	En iyi Sıra	Simpleks İtr. Sayısı	D&S Düğ. Sayısı	CPU (sn)
1	0,4	882	1-8-2-3-6-5-7-9-4	389890	88114	37,4844
	0,6	733	4-1-8-2-3-6-5-7-9	264878	39931	23,1406
	0,8	725	4-1-8-2-3-6-5-7-9	321801	50385	23,5469
2	0,4	1100	1-2-4-5-3-8-7-6-9	241236	51397	22,2812
	0,6	948	6-9-2-4-5-3-8-7-1	303132	50727	25,7812
	0,8	937	6-1-2-4-5-3-8-7-9	317581	49607	23,2812
3	0,4	1232	4-8-1-9-2-3-5-6-7	322333	71742	31,5625
	0,6	1056	7-4-8-1-9-2-3-5-6	403807	72330	36,2188
	0,8	1018	4-7-2-8-1-9-3-5-6	409671	64652	28,8906
4	0,4	1161	2-6-4-9-5-8-7-1-3	642687	184959	71,8438
	0,6	1033	8-2-6-4-9-5-7-1-3	629309	144356	70,2969
	0,8	1021	3-2-6-4-9-5-8-7-1	627673	142991	54,4219
5	0,4	1128	7-2-6-4-5-9-8-1-3	294789	67400	28,3906
	0,6	970	1-8-2-9-5-4-6-7-3	330951	56255	27,6875
	0,8	927	3-1-2-6-4-5-9-8-7	294549	49642	20,5625
6	0,4	1007	5-8-3-7-1-6-2-9-4	202868	45368	19,5469
	0,6	870	9-5-8-3-7-1-6-2-4	262657	48263	24,0312
	0,8	836	2-5-8-3-7-1-6-9-4	413541	69925	34,8281
7	0,4	1281	9-7-3-4-5-1-8-2-6	452024	118198	48,6875
	0,6	1135	8-9-7-3-4-5-1-2-6	593195	129765	60,2812
	0,8	1120	6-2-7-3-4-5-1-8-9	519894	104543	38,9844
8	0,4	1686	9-3-8-4-7-1-6-2-5	734080	217693	83,5156
	0,6	1469	9-5-3-8-4-7-1-6-2	644208	143065	67,25
	0,8	1469	9-5-3-8-4-7-1-6-2	642592	149545	54,1875
9	0,4	1129	6-8-1-7-9-5-4-3-2	92821	15567	7,59375
	0,6	963	2-6-8-7-1-9-5-4-3	210633	28225	15,4844
	0,8	957	2-6-8-1-7-9-5-4-3	216230	28121	14,0312
10	0,4	708	4-5-6-9-3-2-7-1-8	133980	23197	11,0312
	0,6	675	7-4-5-6-9-3-2-1-8	312013	53619	25,1719
	0,8	613	8-4-5-6-9-3-2-7-1	218572	31273	15,5156
11	0,4	1287	1-7-6-3-2-9-5-4-8	267283	60326	25,875
	0,6	1112	4-1-7-6-3-2-9-5-8	300852	51122	26,3125
	0,8	1096	8-1-7-6-3-2-9-5-4	358283	58480	25,0156
12	0,4	807	3-4-7-8-2-1-9-5-6	116143	23292	10,625
	0,6	711	6-3-2-1-8-7-4-9-5	190502	28748	15,9844
	0,8	684	6-3-4-2-1-8-7-9-5	180468	24159	14,1094
13	0,4	944	6-1-2-8-4-5-7-9-3	167968	32359	14,9375
	0,6	779	3-6-1-2-8-4-5-7-9	180351	22535	13,9844
	0,8	773	3-6-1-2-8-4-5-7-9	224929	31059	17,3594
14	0,4	1360	1-9-8-3-6-2-4-7-5	264760	60699	25,5625
	0,6	1137	7-5-9-8-3-6-2-4-1	372102	63001	31,1562
	0,8	1090	7-5-9-2-8-3-6-1-4	402482	68022	27,7812
15	0,4	921	4-2-1-6-3-8-7-5-9	261788	55428	23,8438
	0,6	832	5-4-2-1-6-3-8-7-9	254180	41699	21,8125
	0,8	774	9-4-2-1-6-3-8-7-5	223072	34993	16,9531
Ortalama				337972	67706,16	30,152

EK-2. (Devam) KTP ile bulunan en iyi çözümler ($n=11$)

Problem No	h	En iyi Değer	En iyi Sıra	Simpleks İtr. Sayısı	D&S Düg. Sayısı	CPU (sn)
1	0,4	1946	4-8-10-1-9-3-2-5-11-6-7	13483898	3680393	2257,55
	0,6	1704	4-7-10-1-8-9-3-2-5-11-6	16225940	3629385	2323,06
	0,8	1698	4-7-10-1-8-3-9-2-5-11-6	12504878	2644435	1520,52
2	0,4	1800	8-4-1-9-7-5-11-2-6-3-10	24010618	7563923	4621,8
	0,6	1569	10-3-4-1-9-7-5-11-2-6-8	21464400	5669639	3641,23
	0,8	1545	10-3-4-1-9-7-5-11-2-6-8	22550639	6012432	3007,8
3	0,4	1298	2-8-4-10-5-11-6-9-7-3-1	3902098	711663	469,141
	0,6	1111	1-2-8-4-10-5-11-6-9-7-3	6676800	1057350	778,875
	0,8	1101	1-3-8-4-10-5-11-6-9-2-7	7628761	1145142	691,578
4	0,4	1386	5-11-10-6-7-1-3-9-8-4-2	9166014	2181126	1399,69
	0,6	1209	4-5-11-10-6-7-1-3-9-8-2	12043967	2400016	1726,78
	0,8	1195	4-5-11-10-9-6-7-1-3-2-8	11649564	2340234	1249,16
5	0,4	1844	7-2-9-5-11-3-4-1-10-8-6	28999138	9100080	5884,12
	0,6	1629	8-9-1-10-3-5-11-4-7-2-6	31338544	9833516	5601,2
	0,8	1613	6-2-1-10-3-5-11-4-7-9-8	25346511	7586754	3852,14
6	0,4	2491	8-5-1-4-6-10-3-7-9-2-11	28495673	9515733	5406,06
	0,6	2191	2-11-8-5-1-4-6-10-3-7-9	20317106	6198001	3951,89
	0,8	2191	2-11-8-5-1-4-6-10-3-7-9	18682354	5702300	2820,38
7	0,4	1731	10-6-1-5-7-11-9-8-4-3-2	12960155	3225548	1981,31
	0,6	1538	10-8-4-6-1-9-7-11-5-2-3	12408181	2563485	1758,7
	0,8	1537	2-10-8-4-6-1-9-7-11-5-3	14593298	3032055	1685,75
8	0,4	1063	10-6-11-2-3-5-9-7-4-1-8	8912066	1783860	1199,8
	0,6	915	10-3-1-5-11-2-3-9-4-7-8	4554515	637651	550,172
	0,8	915	10-6-1-5-11-2-3-9-4-7-8	4574313	612734	396
9	0,4	1564	8-11-10-3-6-2-9-4-1-5-7	9632541	2278199	1440,3
	0,6	1372	8-4-11-9-10-3-6-2-7-1-5	9815912	1737989	1235,42
	0,8	1372	8-4-11-9-10-3-6-2-7-1-5	11123913	2079609	1115,55
10	0,4	1256	11-4-7-2-8-1-10-9-5-3-6	15312279	4068771	2558,08
	0,6	1100	6-11-4-7-2-8-1-10-9-5-3	15920740	3362609	2314,73
	0,8	1084	6-11-4-7-2-8-1-10-9-5-3	14536384	2960464	1726,36
11	0,4	1313	9-10-8-2-4-11-5-7-1-3-6	11341922	2761251	1864,27
	0,6	1149	3-7-10-8-2-4-11-5-1-9-6	15113173	3212046	2180,39
	0,8	1120	6-9-1-10-8-2-4-11-5-7-3	14026990	2907103	1697,55
12	0,4	1901	9-10-6-2-3-8-1-11-7-4-5	13925764	3479354	2294,09
	0,6	1683	7-4-10-6-2-3-8-1-11-9-5	10121959	1969709	1397,28
	0,8	1671	5-7-9-10-6-2-3-8-1-11-4	9852829	2040114	1097,59
13	0,4	1462	5-8-11-1-6-4-7-3-2-10-9	14640391	3698493	2481,62
	0,6	1308	10-2-4-7-3-1-8-11-6-5-9	15283519	3245392	2397,95
	0,8	1255	9-10-2-4-7-3-1-8-11-6-5	15038572	2819870	1698,8
14	0,4	1058	1-2-9-11-4-5-8-6-3-7-10	7447211	1741778	1076,72
	0,6	909	10-2-9-5-11-4-8-6-1-3-7	8497832	1407204	1040,78
	0,8	872	10-7-3-5-11-4-9-8-6-1-2	7848028	1155003	754,812
15	0,4	1863	4-1-2-7-9-8-11-10-5-3-6	13085700	3642687	2423,02
	0,6	1619	5-6-2-11-7-9-8-10-3-1-4	14885705	3464630	2363,48
	0,8	1586	6-4-1-2-11-7-9-8-10-5-3	8296364	1667322	969,562
Ortalama				13960826	3433934,49	2108,957

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÖZDEMİR, Müge Hanım
Uyruğu : T.C.
Doğum tarihi ve yeri : 25.08.1982 Emirdağ/AFYONKARAHİSAR
Medeni hali : Bekar
Telefon : 0 (312) 280 26 65
GSM : 0 (505) 438 20 18
e-mail : mgzdmr@yahoo.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi /Endüstri Mühendisliği	2007
Lisans	Gazi Üniversitesi/ Endüstri Mühendisliği	2004
Lise	Fatih Fen Lisesi	2000

Yabancı Dil

İngilizce, Almanca

Hobiler

Sinema, Kitap Okumak, Yüzmek, Kamera ve Fotoğraf Çekimi Yapmak