

**İşlerin Bölünebilir Olduğu
Paralel Makine Çizelgeleme Problemi İçin
Tabu Arama Yöntemi**

Cenk ÇELİK

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Haziran 2008

**Tabu Search Method for a Parallel Machine Scheduling Problem
With a Job Splitting Property**

Cenk ÇELİK

MASTER OF SCIENCE THESIS

Department of Industrial Engineering

June 2008

**İşlerin Bölünebilir Olduğu
Paralel Makine Çizelgeleme Problemi İçin
Tabu Arama Yöntemi**

Cenk ÇELİK

**Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Endüstri Mühendisliği Anabilim Dalı
Endüstri Mühendisliği Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır**

Danışman: Yrd. Doç. Dr. İnci SARIÇİÇEK

Haziran 2008

ONAY

Endüstri Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Cenk Çelik'in YÜKSEK LİSANS tezi olarak hazırladığı “İşlerin Bölünebilir Olduğu Paralel Makine Çizelgeleme Problemi İçin Tabu Arama Yöntemi” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman : Yrd. Doç. Dr. İnci SARIÇİÇEK

İkinci Danışman : -

Yüksek Lisans Tez Savunma Jürisi:

Üye : Yrd. Doç. Dr. İnci SARIÇİÇEK

Üye : Prof. Dr. Nihat YÜZÜGÜLLÜ

Üye : Doç. Dr. Osman PARLAKTUNA

Üye : Yrd. Doç. Dr. Aydın SİPAHİOĞLU

Üye : Yrd. Doç. Dr. Servet HASGÜL

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Çalışmada, işlerin toplam gecikmesinin en küçüklemesi amacıyla, işlerin bölünebilir özelliğe sahip olduğu n bağımsız işli, m özdeş paralel makine çizelgeleme problemi ele alınmıştır. Bu problemde bir iş, alt işlere ayrılabilen ve bu alt işler, paralel makinelerde birbirinden bağımsız olarak işlenebilmektedir. Problem için öncelikle karma tamsayılı bir programlama modeli kurulmuş ve problemin çözümüne bir tabu arama algoritması önerilmiştir. Deney tasarımı ile en iyi parametre seti belirlenen algoritmanın performansı, rassal olarak türetilmiş test problemleri üzerinde test edilmiştir. Lingo yazılımı yardımıyla elde edilen matematiksel modelin sonuçları, önerilen Tabu Aramanın sonuçlarıyla karşılaştırılmıştır. Makul süre zarfında önerilen Tabu Aramanın daha iyi sonuçlar verdiği gözlenmiştir.

Anahtar Kelimeler: Çizelgeleme, paralel makineler, toplam gecikme, iş bölünmesi, tabu arama.

SUMMARY

In this study, we focus on the problem of scheduling n independent jobs on m identical parallel machines with the objective of minimizing total tardiness of the jobs considering a job splitting property. In this problem, it is assumed that a job can be split into sub-jobs and these sub-jobs can be processed independently on parallel machines. This type of problem is formulated as a mixed integer programming model and solved by a tabu search algorithm. The performance of the algorithm, for which the best parameter set is determined by factorial design analysis, is tested on random generated problems with different sizes. The results are compared with those of the mathematical model. Results of the experiments show that the suggested Tabu Search algorithm gives better solutions to the problems in a reasonable amount of computation time.

Keywords: Scheduling, parallel machines, total tardiness, job splitting, tabu search.

TEŐEKKÜR

Tez alıřmalarında, bana danıřmanlık ederek, beni ynlendiren ve her trl olanađı sađlayan danıřmanım Sayın Yrd. Do. Dr. İnci SARIİEK'e teőekkrlerimi sunarım. Ayrıca; yardımları iin Sayın Yrd. Do. Dr. Aydın SİPAHİOĐLU'na teőekkr bir bor bilirim.

Her daim yanımda olan, maddi, manevi hibir fedakarlıktan kaınmayan aileme minnettarlıđımı ifade ederim.

İÇİNDEKİLER

| | <u>Sayfa</u> |
|--|--------------|
| ÖZET | v |
| SUMMARY | vi |
| TEŞEKKÜR | vii |
| ŞEKİLLER DİZİNİ..... | viii |
| ÇİZELGELER DİZİNİ..... | ix |
| SİMGELER VE KISALTMALAR DİZİNİ | x |
| | |
| 1. GİRİŞ | 1 |
| | |
| 2. PARALEL MAKİNE ÇİZELGELEME PROBLEMLERİ VE MEVCUT ÇALIŞMALAR | 4 |
| 2.1. Paralel Makine Çizelgelemenin Önemi | 4 |
| 2.2. Tanımlamalar ve Gösterimler | 4 |
| 2.3. Paralel Makine Çizelgeleme Problemlerinin Sınıflandırılması | 9 |
| 2.3.1. Partiye bağlı olmayan, sıradan bağımsız hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar..... | 9 |
| 2.3.2. Partiye bağlı olmayan, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar..... | 13 |
| 2.3.3. Partiye bağlı, sıradan bağımsız hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar..... | 20 |
| 2.3.4. Partiye bağlı, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar..... | 25 |
| | |
| 3. İŞLERİN BÖLÜNEBİLİR OLDUĞU, SIRADAN BAĞIMSIZ HAZIRLIK SÜRELERİNİ İÇEREN ÖZDEŞ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ..... | 29 |
| 3.1. Çizelgelemenin Tanımı ve Çizelgeleme Problemi | 29 |
| 3.2. Bir Örgüt İçerisinde Çizelgelemenin Yeri Ve Önemi..... | 31 |
| 3.3. Problemin Tanımlanması | 34 |

İÇİNDEKİLER (devam)

| | <u>Sayfa</u> |
|--|--------------|
| 3.4. Varsayımlar..... | 35 |
| 3.5. Terminoloji | 36 |
| 3.6. Probleme İlişkin Matematiksel Model..... | 36 |
| 3.7. Problemin Zorluğu | 40 |
| 4. İŞLERİN BÖLÜNEBİLİR OLDUĞU, SIRADAN BAĞIMSIZ HAZIRLIK SÜRELERİNİ İÇEREN ÖZDEŞ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ İÇİN TABU ARAMA YÖNTEMİ..... | 41 |
| 4.1. Tanım | 41 |
| 4.2. Tabu Aramanın Esasları..... | 41 |
| 4.3. Kısa Dönem Hafıza..... | 43 |
| 4.3.1. Komşuluk ve aday liste stratejileri..... | 45 |
| 4.3.2. Tabu listesi | 47 |
| 4.3.3. Tabu yıkma kriteri..... | 48 |
| 4.3.4. Tabu arama algoritması..... | 49 |
| 4.4. Uzun Dönem Hafıza | 51 |
| 4.5. İşlerin Bölünebilir Olduğu, Sıradan Bağımsız Hazırlık Sürelerini İçeren Özdeş Paralel Makine Çizelgeleme Problemi İçin Tabu Arama Yöntemi | 52 |
| 4.6. Deneysel Çalışmalar | 56 |
| 4.6.1. Tabu arama parametrelerinin belirlenmesi | 57 |
| 4.6.2. Önerilen Tabu Aramanın test problemleri üzerindeki performansı..... | 61 |
| 5. SONUÇ VE ÖNERİLER..... | 66 |
| KAYNAKLAR DİZİNİ..... | 68 |
| EKLER | |

ŞEKİLLER DİZİNİ

| <u>Şekil</u> | <u>Sayfa</u> |
|---|---------------------|
| 3.1 Yukarıdan aşağıya planlama | 32 |
| 3.2 Bir imalat sistemindeki bilgi akış diyagramı | 33 |
| 4.1 Kısa dönem hafıza bileşenleri..... | 45 |
| 4.2 En uygun adayın seçimi..... | 46 |
| 4.3 Test Problemleri-I için ana etkiler grafiği..... | 59 |
| 4.4. Test Problemleri-II için ana etkiler grafiği | 61 |

ÇİZELGELER DİZİNİ

| <u>Cizelge</u> | <u>Sayfa</u> |
|---|--------------|
| 2.1 Partiye bağılı olmayan, sıradan bağımsız hazırlık süreli paralel makine çizelgeleme problemlerini içeren çalışmalar | 10 |
| 2.2 Partiye bağılı olmayan, sıraya bağımlı hazırlık süreli paralel makine çizelgeleme problemlerini içeren çalışmalar | 14 |
| 2.3 Partiye bağılı, sıradan bağımsız hazırlık süreli paralel makine çizelgeleme problemlerini içeren çalışmalar | 21 |
| 2.4 Partiye bağılı, sıraya bağımlı hazırlık süreli paralel makine çizelgeleme problemlerini içeren çalışmalar | 26 |
| 4.1. Tabu Aramanın uygulama alanları | 42 |
| 4.2. Örnek probleme dair veriler..... | 50 |
| 4.3. İkili yer değıştirme için nitelikler ve tabu kısıtlamaları..... | 55 |
| 4.4 Test Problemleri-I ve II için deney tasarımıda göz önüne alınan faktör ve düzeyleri..... | 57 |
| 4.5 Test Problemleri-I için Varyans Analizi sonuçları | 59 |
| 4.6 Test Problemleri-II için Varyans Analizi sonuçları | 60 |
| 4.7 Küçük test problemleri üzerinde, önerilen Tabu Aramanın en iyi çözümlerden sapma yüzdeleri | 62 |
| 4.8 Önerilen Tabu Aramanın Test Problemleri-I üzerindeki performansı..... | 63 |
| 4.9 Önerilen Tabu Aramanın Test Problemleri-II üzerindeki performansı | 64 |

SİMGELER VE KISALTMALAR DİZİNİ

| <u>Simgeler</u> | <u>Açıklama</u> |
|------------------------|--|
| P | Özdeş paralel makineler |
| Q | Benzer paralel makineler |
| R | Farklı paralel makineler |
| ST_{si} | Sıradan bağımsız hazırlık süresi |
| SC_{si} | Sıradan bağımsız hazırlık maliyeti |
| ST_{sd} | Sıraya bağımlı hazırlık süresi |
| SC_{sd} | Sıraya bağımlı hazırlık maliyeti |
| $ST_{si,b}$ | Sıradan bağımsız parti hazırlık süresi |
| $SC_{si,b}$ | Sıradan bağımsız parti hazırlık maliyeti |
| $ST_{sd,b}$ | Sıraya bağımlı parti hazırlık süresi |
| $SC_{sd,b}$ | Sıraya bağımlı parti hazırlık maliyeti |
| C_{max} | En büyük tamamlanma zamanı |
| L_{max} | En büyük sapma |
| T_{max} | En büyük gecikme |
| TSC | Toplam hazırlık/geçiş maliyeti |
| TST | Toplam hazırlık/geçiş süresi |
| TCT | Toplam tamamlanma süresi |
| TE | Toplam erkenlik |
| TT | Toplam gecikme |
| MFT | Ortalama akış süresi |
| MCT | Ortalama tamamlanma süresi |
| MT | Ortalama gecikme |
| NLJ | Geciken iş sayısı |
| WTE | Ağırlıklı toplam erkenlik |
| WTT | Ağırlıklı toplam gecikme |
| $WMFT$ | Ağırlıklı ortalama akış süresi |
| $WTFT$ | Ağırlıklı toplam akış süresi |

| | |
|-------------|--------------------------------------|
| <i>WMCT</i> | Ağırlıklı ortalama tamamlanma süresi |
| <i>WTCT</i> | Ağırlıklı toplam tamamlanma süresi |

Kısaltmalar **Açıklama**

| | |
|-----|------------------------|
| GA | Genetik Algoritma |
| SEK | Sevk Etme Kuralı |
| TA | Tabu (Yasaklı) Arama |
| GSP | Gezgin Satıcı Problemi |
| YSA | Yapay Sinir Ağları |

BÖLÜM 1

GİRİŞ

Çizelgeleme, bir çok üretim ve hizmet endüstrisinde önemli rol oynayan bir karar verme süreci olup, tedarikten üretime, taşıma ve dağıtımdan, bilgi işleme ve iletişime bir çok alanda kullanılmaktadır. Çizelgeleme ile bir firmadaki kıt kaynakların matematiksel teknikler veya sezgisel yöntemler kullanılarak, görevlere en iyi şekilde tahsis edilmesi rekabetçi ortamda ayakta kalabilmek için bir gereklilik halini almıştır.

Çizelgeleme konusunda paralel makinelerde çizelgeleme önemli bir alt konu olup, Pinedo'ya (2002) göre paralel makineler, hem teorik hem de pratik açıdan önem arz etmektedir. Teorik açıdan; tek makinenin genel bir hâli ve esnek akış atölyesinin özel bir hâlidir. Pratik açıdan bakıldığında ise; paralel kaynaklara günlük hayatta yaygın olarak karşılaşılmaktadır. Ayrıca paralel makine teknikleri, çok kademeli sistemler için ayrıştırma prosedürleri dâhilinde sıklıkla kullanılmaktadır.

Paralel makine çizelgeleme teorisi, makinelerin işlere dağılımı ve belli bir amaca göre her makinede işlem sırasının belirlenmesi ile ilgilidir. Paralel makine çizelgelemesi, yaygın kullanım alanlarının olmasından dolayı popüler bir araştırma konusu olmuştur ve bu popülerite son yıllarda paralel-işlemcili bilgisayarlar teknolojisinin gelişmesiyle önemli ölçüde artmıştır (Azizoğlu, 1994).

Paralel makinelerde çizelgeleme iki adımda düşünülebilir. Birincisi, işlerin hangi makineye atanması gerektiğinin kararlaştırılması; ikincisi ise her bir makineye atanan işlerin sırasının belirlenmesidir. En büyük tamamlanma zamanı (makespan) açısından yalnızca atama süreci önem arz etmektedir (Pinedo, 2002).

Teslim tarihiyle ilişkili performans ölçütlerini içeren problemler bir çok araştırmaya konu olsa da özdeş paralel makinelerin özel durumları haricinde paralel

makine çizelgeleme problemlerinde toplam gecikmeyi en küçükleyen amaç için fazla ilerleme kaydedilmemiştir (Shim and Kim, 2008). Ayrıca, pratik büyüklükteki paralel makine gecikme problemleri için en iyi sonuç elde etmek kolay olmadığından araştırmacılar sezgisel algoritmalara başvurmuşlardır (Kim et al., 2004).

İşlerin bölünebilir özellikte olduğu paralel makine çizelgelemesi üzerine çok az çalışma mevcuttur. Serafini (1996), her bir işin keyfi bölünebildiği, alt işlerin sayısının kesikli değil de sürekli olarak düşünüldüğü, ayrıca benzer ya da farklı paralel makineler üzerinde bağımsızca işlenebildiği durumları incelemiş ve en büyük ağırlıklı gecikmeyi en küçükleyen amaca bir sezgisel algoritma sunmuştur. Her bir işin keyfi bölünebildiği özdeş işlemcili problem için Xing and Zhang (2000), en büyük tamamlanma zamanını en küçükleyen amaca bir sezgisel algoritma önermiş ve algoritmanın en kötü durum performansını analiz etmiştir. Aynı şekilde Tahar et al.'da (2006) çalışmalarında işlerin keyfi bölünebilir özellikte olduğu durumu ele almış; probleme, doğrusal programlama tabanlı bir sezgiselle çözüm aramıştır.

Kim et al. (2004), her bir işin keyfi olarak bölünmeyip, kesikli sayıda birim işlerden oluştuğu problem yapısı tanımlamıştır. Tanımladıkları probleme iki safhalı bir sezgisel algoritmayla çözüm aramıştır. İlgili probleme dair başka bir sezgisel olmadığı için literatürde var olan sezgisel bir algoritmayı, kendi problemlerine uyarlayarak onun sonuçlarıyla kendilerinininkini karşılaştırmıştır. Daha sonra aynı problem yapısını, Shim and Kim (2008) dal sınır algoritmasıyla çözmeye çalışmışlardır. Dal sınır algoritmasının performansı için çeşitli alt sınırlar ve baskın özellikler (dominance properties) tanımlamışlardır. Çalışmaları sonucunda, dal sınır algoritmasının maksimum dört makine ve on iki iş ile beş makine ve sekiz işe kadar makul zamanda sonuca ulaşabilmişlerdir.

Çalışmada, hem daha önce ilgili probleme meta sezgisel bir yaklaşımın uygulanmamış olması nedeniyle hem de probleme makul zamanda en iyiye yakın çözümler oluşturabilmek adına, çizelgeleme alanında başarılı sonuçlar verdiği bilinen Tabu Aramasıyla çözüm aranmıştır.

Çalışmanın birinci bölümünde konuya giriş yapılmış, ikinci bölümde paralel makine çizelgeleme tanım ve gösterimleri ile literatürdeki çalışmalara yer verilmiştir. Üçüncü bölümde, ele alınan, işlerin bölünebilir olduğu, sıradan bağımsız süreleri içeren özdeş paralel makine problemi için bir karma tamsayılı matematiksel model geliştirilmiştir. Dördüncü bölümde ise önerilen Tabu Aramaya ait parametreler hesaplanmış, Tabu Aramanın performansı rassal olarak oluşturulan çeşitli problem setleri üzerinde denenmiştir.

BÖLÜM 2

PARALEL MAKİNE ÇİZELGELEME PROBLEMLERİ VE MEVCUT ÇALIŞMALAR

2.1. Paralel Makine Çizelgelemenin Önemi

Paralel makine çizelgeleme teorisi, makinelerin işlere dağılımı ve belli bir amaca göre her makinede işlem sırasının belirlenmesi ile ilgilidir. Paralel makine çizelgelemesi, yaygın kullanım alanlarının olmasından dolayı popüler bir araştırma konusu olmuştur ve bu popülerite son yıllarda paralel-işlemcili bilgisayarlar teknolojisinin gelişmesiyle önemli ölçüde artmıştır (Azizoğlu, 1994).

Pinedo'ya (2002) göre paralel makineler, hem teorik hem de pratik açıdan önem arz etmektedir. Teorik açıdan; tek makinenin genel bir hâli ve esnek akış atölyesinin (flexible flow shop) özel bir hâlidir. Pratik açıdan bakıldığında ise; paralel kaynaklara günlük hayatta yaygın olarak karşılaşılmaktadır. Ayrıca paralel makine teknikleri, çok kademeli sistemler için (multi-stage systems) ayrıştırma prosedürleri dâhilinde sıklıkla kullanılmaktadır.

2.2. Tanımlamalar ve Gösterimler

Paralel Makine Çizelgeleme konusundaki mevcut çalışmalar verilmeden önce, literatüre ilişkin bazı önemli tanımlamaların ve gösterimlerin verilmesi yararlı olacaktır.

İşlem süresi (p_{ij}): p_{ij} , j işinin i makinesindeki işlem süresini ifade etmektedir. j işinin işlem süresi makinelerden bağımsız ise ya da j işi, yalnızca belirlenmiş tek bir makine tarafından işlenecekse, i altsimgesi gösterime dâhil edilmez (Pinedo, 2002).

Hazır olma zamanı (r_j): j işinin sisteme geldiği zamanı, j işinin işlenebileceği en erken zamanı, ifade etmektedir (Pinedo, 2002).

Tamamlanma zamanı (C_j): j işinin tamamlanma zamanını ifade eder.

Teslim zamanı (d_j): j işinin müşteriye teslim edileceği zamanı ifade eder. İşlerin teslim zamanlarının kesinlikle karşılanması isteniyor ve teslim zamanlarının ihmaline izin verilmiyorsa son tarih (*deadline*) olarak adlandırılır (Alpay, 2003).

Akış süresi (F_j): j işinin hazır olma zamanı ile tamamlanma zamanı arasında sistemde geçirdiği süreyi ifade eder ve

$$F_j = C_j - r_j \quad (F_j > 0)$$

şeklinde gösterilir (Sipper and Bulfin, 1997).

Sapma (L_j): j işinin tamamlanma zamanı ile teslim zamanı arasındaki farkı ifade eder ve negatif ya da pozitif değer alabilir.

$$L_j = C_j - d_j$$

şeklinde gösterilir (Sipper and Bulfin, 1997).

Erkenlik (E_j): j işinin teslim zamanından önce tamamlanmasının bir göstergesi olup,

$$E_j = \text{enb}\{0, -L_j\}$$

şeklinde ifade edilir (Sipper and Bulfin, 1997).

Gecikme (T_j): j işinin teslim zamanından sonra tamamlanmasının bir göstergesidir ve

$$T_j = \text{enb}\{0, L_j\}$$

şeklinde gösterilir (Sipper and Bulfin, 1997).

Ağırlık (w_j): j işine ait ağırlık, temel olarak sistemde j işinin diğer işlere göre önemini ifade eden bir öncelik faktörüdür. Örneğin; ilgili ağırlık, işi sistemde tutmanın gerçek maliyetini ifade edebilir. İlgili maliyet, elde bulundurma ya da stok maliyeti olabilir; ya da o işe atanan bir değeri ifade edebilir (Pinedo, 2002).

Bir çizelgeleme problemi, $\alpha | \beta | \gamma$ üçlüsüyle tanımlanmaktadır. α alanı makine çevresini ifade etmekte olup tek bir giriş söz konusudur. β alanında, işlem özelliklerine ait ayrıntılar ve kısıtlar tanımlanmaktadır. Bu alana gerekli durumlarda birden fazla giriş söz konusu olacağı gibi hiçbir giriş de yapılmayabilir. γ alanında ise performans ölçütü yer almaktadır ve genellikle tek bir giriş söz konusudur (Pinedo, 2002).

Paralel makine çizelgeleme çalışmalarında α alanında yapılan tanımlamalar şu şekildedir:

Özdeş paralel makineler (*identical parallel machines, P_m*): Paralel m adet makine söz konusudur. j işi tek bir işlem gerektirmekte ve ilgili m sayıdaki makinelerin ya da belirli bir altkümeye ait makinelerin herhangi birinde işlenebilmektedir. Şayet j işi belirli bir M_j kümesine ait makinelerin haricinde diğer herhangi bir makinede işlenmesine izin verilmiyorsa, β alanında da M_j ibaresi yer almalıdır (Pinedo, 2002).

Farklı hızlara sahip paralel makineler (*uniform paralel machines, Q_m*): Farklı hızlara sahip paralel m sayıda makine söz konusudur. i makinesine ait hız, v_i ile gösterilmektedir. j işinin i makinesindeki işlem süresi olan p_{ij} , p_j/v_i 'ye eşittir (j işinin tüm işleminin i makinesi tarafından görüldüğü farz edilerek). Bu, *benzer* (*uniform*) makineler olarak ifade edilir. Şayet tüm makineler aynı hıza sahipse, yani tüm i 'ler için $v_i=1$ ve $p_{ij}=p_j$ şeklinde ise, bir önceki özdeşlik söz konusu olur (Pinedo, 2002).

Farklı paralel makineler (*unrelated paralel machines, R_m*): Bu durum bir öncekinin genel bir hâlidir. Paralel m sayıda farklı makine söz konusudur. i makinesi j işini v_{ij} hızıyla işlemektedir. j işinin i makinesindeki işlem süresi olan p_{ij} , p_j/v_{ij} 'ye eşittir (j işinin tüm işleminin i makinesi tarafından görüldüğü farz edilerek). Şayet makine hızları işlerden bağımsızsa, yani tüm i ve j 'ler için $v_{ij}=v_i$ ise, bir önceki durum söz konusu olur (Pinedo, 2002).

β alanında yer alan işlem kısıtlarına birden çok giriş yapılabilir. Bu alana yapılan olası girişler şöyledir (Pinedo, 2002):

Hazır olma zamanı (r_j): β alanında bu sembol mevcutsa, j işine ait bu zaman gelmeden önce işlemine başlanamaz. Aksi hâlde herhangi bir zamanda j işine başlanılabilir. Hazır olma zamanlarının aksine, teslim zamanları bu alanda gösterilmemektedir. Teslim zamanlarının varlığına amaç fonksiyonunun yapısı karar vermektedir.

Sıraya bağımlı hazırlık süreleri (s_{jk}): j ve k işleri arasındaki sıraya bağımlı hazırlık süresini ifade etmektedir. s_{0k} , sıralamadaki ilk iş olan k işinin hazırlık süresini; s_{j0} ise, sıralamadaki en son iş olan j işinden sonraki temizlik süresini göstermektedir. Tabii ki; bu değerler sıfır olabilir. Şayet j ve k işleri arasındaki hazırlık süresi makineye de bağlıysa gösterime i altsimgesi de eklenmektedir (yani gösterim s_{ijk} hâlini almaktadır). β alanında herhangi bir s_{jk} gösterimi mevcut değilse, tüm hazırlık süreleri sıfır olarak ya da sıradan bağımsız olarak, basit hâliyle işlem süresine dâhil edilmiş olarak, kabul edilmektedir.

İş kesintileri ($prmp$): İş kesintileri, bir işin başlangıcından tamamlanıncaya kadar tek bir makine tarafından tutulmasının gerek olmadığını ifade eder. Çizelgeci herhangi bir anda bir işin işlenmesine ara verebilir ve o makineye farklı bir iş yükleyebilir. Kesintiye uğramış işin işlenme miktarında bir kayıp söz konusu olmaz. Kesintiye uğramış bir iş tekrardan makineye yüklendiğinde ya da paralel makine durumundaki bir diğer makineye yüklendiğinde, yalnızca kalan işlem süresi kadar makinede işlenmeye ihtiyaç duyar. İş kesintilerine izin verildiği durumda, β alanına $prmp$ ifadesi eklenir. Şayet $prmp$ ifadesi yer almıyorsa, iş kesintisine izin verilmediği anlaşılmalıdır.

İş bölünmesi ($split$): Herhangi bir işin bölünebildiği ve ilgili işe ait bölünen parçaların eş zamanlı olarak, birbirlerinden bağımsızca farklı makinelerde işlenebildiği durumu göstermektedir. İş bölünmesine izin verildiği durumda, β alanına $split$ ifadesi eklenir. Şayet $split$ ifadesi yer almıyorsa, iş bölünmesine izin verilmediği anlaşılmalıdır.

İş öncelik kısıtları ($prec$): İş öncelik kısıtları, bir işin işlenmeye başlanmasından önce bir ya da daha fazla işin tamamlanmış olmasını gerektirir. İş öncelik kısıtları bir çok şekilde karşımıza çıkmaktadır. Bu kısıtlar, her bir işin en fazla bir öncülü ve ardılı varsa, zincirler olarak; her bir işin en fazla bir ardılı varsa, *intree* olarak; her bir işin en fazla bir öncülü varsa da *outtree* olarak nitelendirilmektedirler. β alanına $prec$ ifadesi yer almıyorsa, işlerin herhangi bir iş öncelik kısıtlarına tabi olmadığı anlaşılmaktadır.

Uygun makine kısıtları (M_j): j işini işleyebilen makinelerin kümesini ifade eder. Şayet β alanında bu ifade yer almıyorsa, j işi m sayıdaki makinelerin herhangi birinde işlenebilir demektir.

Performans ölçütünün yer aldığı γ alanına en küçüklenmek amacıyla yapılan olası girişler ise şöyledir (Pinedo, 2002):

En büyük tamamlanma zamanı (C_{max}): En büyük tamamlanma zamanı $\max(C_1, \dots, C_n)$ şeklinde tanımlanmakta olup sistemi terk eden en son işe ait tamamlanma zamanını ifade etmektedir.

En büyük sapma (L_{max}): En büyük sapma, $\max(L_1, \dots, L_n)$ olarak tanımlanmaktadır. En kötü teslim zamanı ihlalini göstermektedir.

Toplam ağırlıklı tamamlanma zamanı ($\sum w_j C_j$): n tane işe ait ağırlıklı tamamlanma zamanlarının toplamı, çizelgeye ait toplam elde bulundurma ya da stok maliyetlerine dair bir gösterge sunmaktadır.

Ağırlıklı geciken işlerin sayısı ($\sum w_j U_j$): Toplam ağırlıklandırılmış geciken iş sayısını ifade etmektedir. Kolaylıkla kaydedilebilirliğinin yanında pratikte sıklıkla kullanılan bir ölçüttür.

Toplam Gecikme ($\sum T_j$): Her bir işe ait gecikmelerin (T_j) toplamıdır. İşlerin, teslim zamanlarından önce tamamlanmasının bir yarar sağlamadığı durumlarda kullanılmaktadır.

2.3. Paralel Makine Çizelgeleme Problemlerinin Sınıflandırılması

Allahverdi et al. (1999, 2008) çalışmalarında paralel makine çizelgeleme problemlerini öncelikle *partiye bağlı hazırlık süreli/maliyetli* (batching setup times/costs) ve *partiye bağlı olmayan hazırlık süreli/maliyetli* (non-batching setup times/costs) olarak sınıflandırmıştır. Daha sonra bu sınıfları kendi içlerinde *sıradan bağımsız hazırlık süreli/maliyetli* (sequence-independent setup) problemler ve *sıraya bağımlı hazırlık süreli/maliyetli* (sequence-dependent setup) problemler olmak üzere alt sınıflara ayırmıştır.

2.3.1. Partiye bağlı olmayan, sıradan bağımsız hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar

Partiye bağlı olmayan, sıradan bağımsız hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar Çizelge 2.1'de verilmiştir. Bu alandaki çalışmalar daha çok karmaşıklık hesaplamalarını ve özel durumlar için geliştirilen polinom ya da sözde-polinom algoritmaları içermektedir.

Xing and Zhang (2000), bir işin aynı anda farklı iki makinede işlenebildiği durumu göz önüne almıştır. Bu problem için Xing and Zhang (2000), en büyük tamamlanma süresini en küçükleme amacıyla, sıradan bağımsız hazırlık sürelerini içeren bir probleme en kötü durum performans oranı (worst-case performance ratio) ile birlikte sezgisel bir yöntem sunmuşlardır. İşlerin kesintiye uğradığı her anda hazırlık süresinin gerekliliği varsayımı altında Schuurman and Woeginger (1999), $P|ST_{si},pmtn|C_{max}$ problemine bir yaklaşım algoritması (approximation algorithm) önermişlerdir. Ayrıca; hazırlık sürelerinin eşit olduğu durumda bu problemi çözen polinom zamanlı bir algoritmanın varlığını ispat etmişlerdir (Allahverdi et al., 2008).

Çizelge 2.1. Partiye bağlı olmayan, sıradan bağımsız hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar.

| Yazarlar | Paralel Makine Yapısı Açısından | | | Amaç Fonksiyonu Açısından | | Yaklaşım |
|--------------------------------|---------------------------------|---|---|---|-----------------------------------|---|
| | P | Q | R | Teslim Zamanıyla İlişkili | Diğer | |
| Koulamas (1996) | √ | | | | Makine aylak süresi | Demet Araması Sezgiseli |
| Kravchenko and Werner (1997) | √ | | | | Cmax | Liste Çizelgeleme Sezgiselleri |
| Kravchenko and Werner (1998) | √ | | | | Cmax | Sözde polinom algoritması |
| Schuurman and Woeginger (1999) | √ | | | | Cmax | Algoritma, En kötü durum oranı |
| Glass et al. (2000) | √ | | | | Cmax | Algoritmalar, En kötü durum oranı |
| Hall et al. (2000) | √ | | | $L_{max}, \sum T_j, \sum w_j T_j, \sum U_j, \sum w_j U_j$ | $C_{max}, \sum C_j, \sum w_j C_j$ | Polinom ve Sözde Polinom Algoritmalar |
| Xing and Zhang (2000) | √ | | | | Cmax | Sezgisel, En kötü durum performans oranı |
| Kravchenko and Werner (2001) | √ | | | | $\sum C_j$ | Sezgisel, hata sınırı |
| Wang and Cheng (2001) | √ | | | | $\sum w_j C_j$ | Yaklaşım Algoritması, En kötü durum performansı |
| Abdekhodae and Wirth (2002) | √ | | | | Cmax | Tamsayılı Programlama, Polinom Algoritmalar |
| Brucker et al. (2002) | √ | | | $L_{max}, \sum T_j, \sum w_j T_j, \sum U_j, \sum w_j U_j$ | $C_{max}, \sum C_j, \sum w_j C_j$ | Polinom Algoritmalar |
| Abdekhodae et al. (2004) | √ | | | | Cmax | Sezgiseller, Alt Sınır |
| Guirchoun et al. (2005) | √ | | | $L_{max}, \sum T_j, \sum w_j T_j, \sum U_j, \sum w_j U_j$ | $C_{max}, \sum C_j, \sum w_j C_j$ | Karmaşıklık sonuçları |
| Abdekhodae et al. (2006) | √ | | | | Cmax | Aç Gözlü Sezgisel, Genetik Algoritma |

İzleyen paralel makine çizelgeleme problemi literatürde son zamanlarda yer almaktadır. Bir işin bir makineye yüklenmesi, hazırlık süresi olarak adlandırılan süre, tek bir sunucu (server) tarafından gerçekleştirilmektedir. Bir makine bir işi işlerken, bu hazırlık süresi uygulanamamaktadır. Diğer yandan, iş makineye yüklendikten sonra sunucu olmaksızın makine ilgili işi işleyebilmektedir. Makineler tarafından sunucuya olan eş zamanlı istekler, makine aylak süresi (machine idle time) olarak değerlendirilmektedir. Bu problem, $P,S|ST_{si}|\gamma$ olarak gösterilmektedir. Her işe ait hazırlık süreleri birbirine eşitse, $ST_{si} = s$ olarak gösterilmektedir (Allahverdi et al., 2008).

Kravchenko and Werner (1997), $P,S|ST_{si}=s|C_{max}$ problemi üzerine çalışmışlar ve problemin üst düzey (strongly) NP-zor sınıfına girdiğini göstermişlerdir. Problem üzerinde bazı liste çizelgeleme sezgisellerini (list scheduling heuristics) analiz etmişler ve polinom zamanda çözülen bazı durumları göstermişlerdir. Kravchenko and Werner (2001), aynı problemi birim hazırlık süreli ve toplam tamamlanma zamanı ölçütü altında (yani $P,S|ST_{si}=1|\sum C_j$) inceleyip, problemi çözen bir sezgisel algoritma önermişlerdir. Wang and Cheng (2001), $P,S|ST_{si}|\sum w_j C_j$ problemini ele almışlar ve bir $(5-1/m)$ yaklaşım algoritması sunmuşlardır. Ayrıca; En Kısa İşlem Süresi (Shortest Processing Time-SPT) çizelgelemesinin $P,S|ST_{si}=s|\sum C_j$ problemine $3/2$ yaklaşımı olduğunu göstermişlerdir (Allahverdi et al., 2008).

Problem ayrıca iki paralel makine durumu olarak da incelenmiştir. Koulamas (1996), sunucunun elde bulundurulamamasından kaynaklanan makine aylak süresini en küçükleyen $P2,S|ST_{si}$ probleminin üst düzey NP-zor sınıfına girdiğini göstermiştir. Probleme *etkili demet araması sezgiseli* (efficient beam search heuristic) önermiştir. Abdekhodae and Wirth (2002), *alternating job processing* adında tanımladıkları bir varsayım altında $P2,S|ST_{si}|C_{max}$ problemini ele almışlardır. Problemin üst düzey NP-zor sınıfında olduğu kanıtlamışlar, bir tamsayı programlama formülasyonu önermişler ve bir çok kısıtlayıcı durum için polinom zamanlı algoritmalar sunmuşlardır. Abdekhodae et al. (2004), aynı problemi eşit işlem ve hazırlık süreli olarak incelemişlerdir. Bu özel durumlara sahip aynı problemin NP-zor sınıfına girdiğini göstermişler ve her bir durum için sezgiseller önermişlerdir. Abdekhodae et al. (2006) ayrıca aynı problemi genel

durum için de incelemişlerdir. Genel durum için, aç gözlü sezgiseller ve bir genetik algoritma önermişlerdir. Aynı zamanda genel durumlu bu problemi çözmek için, bilinen Gilmore-Gomory algoritmasını da önermişlerdir (Allahverdi et al., 2008).

Hall et al. (2000), hazırlık sürelerinin eşit olduğu, C_{max} ve $\sum C_j$ amaçlarını ele alan $P2, S|ST_{si}$ problemlerinin üst düzey NP-zor sınıfına girdiğini kanıtlamışlardır. Şayet tüm işlerin işlem süreleri bir birimse, $\sum T_j$ ve $\sum w_j U_j$ amaçlarını içeren $P2, S|ST_{si}$ problemi, NP-zor; $\sum w_j T_j$ amacını içeren problem, üst düzey NP-zor sınıfına girmekte; $C_{max}, L_{max}, \sum C_j, \sum w_j C_j$ ve $\sum U_j$ amaçlarını içeren $P, S|ST_{si}$ problemi ise polinom zamanda çözülebilmektedir. Brucker et al. (2002), $P, S|ST_{si}|\sum C_j$ probleminin üst düzey NP-zor sınıfına girdiğini kanıtlamıştır. Brucker et al., paralel makineleri içeren sunucu çizelgeleme problemleri için sayısız karmaşıklık (complexity) sonuçları elde etmiştir. Birim hazırlık süreli $P3, S|ST_{si}|\sum C_j$ problemi için bir $O(n^7)$ algoritması ve eşit hazırlık ve işlem süresi gibi özel durumlara ait birçok polinom zamanlı algoritmalar geliştirmişlerdir (Allahverdi et al., 2008).

Diğer bir ayrıntılı çalışma, tahsisli makine durumunu (*dedicated machines*, her bir makinenin kendisine önceden atanmış iş kümelerini işleyebildiği durum) ve C_{max} amacını içeren problemin Glass et al. (2000) tarafından yapılmıştır. Diğer bir deyişle; n sayıdaki işler, bir kümedeki işlerin aynı makinede işlendiği m sayıda kümeye ayrılmaktadır. İki tahsisli makineyi içeren problemin, hazırlık sürelerinin eşit olduğu ya da işlem sürelerinin eşit olduğu durumda bile üst düzey NP-zor sınıfına dâhil olduğunu kanıtlamışlardır. Basit bir aç gözlü algoritmanın m makineli durum için en iyi değerinin en fazla iki katı bir çizelge ürettiğini kanıtlamışlardır. Ayrıca; iki makineli durum için kötü durum oranının $3/2$ olduğu bir sezgisel sunmuşlardır (Allahverdi et al., 2008).

Yukarıda bahsedilen tüm çalışmalarda, hazırlık sürelerinin tek bir sunucu tarafından gerçekleştirildiği farz edilmiştir. Kravchenko and Werner (1998), m makinenin olduğu $m-1$ sunuculu problemleri ele almışlardır. Amacın en büyük tamamlanma zamanını en küçükleme olduğu problem için sözde polinom zamanlı bir algoritma sunmuşlardır (Allahverdi et al., 2008).

2.3.2. Partiye bağı olmayan, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar

Partiye bağı olmayan, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar Çizelge 2.2’de verilmiştir. En kapsamlı Paralel Makine Çizelgeleme problemlerini içeren çalışmalar bu alanda yapılmıştır. Çalışmalar daha çok, aynı probleme ait önceki çalışmalarda elde edilen sonuçlardan daha iyi sonuçlar veren yöntemleri içermektedir.

Marsh and Montgomery (1973), hazırlık sürelerini en küçükleme temelinde makinelerin özdeş ve özdeş olmadığı iki durum göz önüne alınarak sıra bağımlı hazırlık sürelerini içeren paralel makine problemine sezgisel bir yaklaşım önermişlerdir. Makinelerin özdeş olmadığı durumda, geçiş süreleri (changeover times) hem sıralamaya hem de makineye bağıdır. Deane and White (1975), Marsh ve Montgomery’nin çalışmasını iş yükü dengelemeyi (workload balancing) de ekleyerek genişletmişlerdir. Sıra bağımlı hazırlık maliyetini en küçükleme için bir dal-sınır algoritması önermişlerdir. Geoffrion and Graves (1976), geçiş (changeover), üretim ve zaman kısıtlı ceza maliyetlerinin toplamını en küçükleme için probleme bir ikinci dereceden atama formülasyonu (quadratic assignment formulation) geliştirmişlerdir. Guinet (1990), $R|ST_{sd}|TFT$ ve $R|ST_{sd}|T_{max}$ problemlerine sezgisel algoritmalar geliştirmek için karma tamsayı programlama formülasyonu (mixed integer programming formulation) kullanmıştır. Elmaghraby et al. (1993), Guinet’in (1990) çalışmasını genişleterek kapsamlı yinelemeli sezgiselini (extended iterative heuristic) geliştirmiştir (Allahverdi et al., 1999).

Çizelge 2.2. Partiye bağlı olmayan, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar.

| Yazarlar | Paralel Makine Yapısı Açısından | | | Amaç Fonksiyonu Açısından | | Yaklaşım |
|---------------------------------------|---------------------------------|---|---|--------------------------------------|---------------|---|
| | P | Q | R | Teslim Zamanıyla İlişkili | Diğer | |
| Marsh and Montgomery (1973) | √ | | √ | | TSC | Sezgisel |
| Deane and White (1975) | √ | | | | TSC | Dal-Sınır Algoritması |
| Geoffrion and Graves (1976) | √ | | | | Maliyet Fonk. | 2. Dereceden Atama Formülasyonu |
| Guinet (1990) | | | √ | MT | MCT | Karma Tamsayılı Programlama |
| Elmaghraby (1993) | | | √ | MT | MCT | Yinelemeli Sezgisel |
| Parker et al. (1977) | √ | | | | TSC | Araç Rotalama Problemi, Algoritmalar |
| Dearing and Henderson (1984) | √ | | | | TSC | Tamsayılı Doğrusal Programlama |
| Sumichrast and Baker (1987) | √ | | | | TST | Tamsayılı Programlama Temelli Sezgisel |
| Ovacik and Uzsoy (1993) | √ | | | Lmax | Cmax | Sezgiseller |
| Franca et al. (1996) | √ | | | | Cmax | Yasaklı Aramanın yer aldığı 3 Safhalı Sezgisel |
| Ovacik and Uzsoy (1995) | √ | | | Lmax | | Family of Rolling Horizon isimli Sezgiseller |
| Guinet and Dussauchoy (1993) | √ | | | | Cmax | Genişletilmiş Macar Algoritması |
| Lee and Pinedo (1997) | √ | | | WTT | | SEK ve Tavlama Benzetimini içeren Üç-Safhalı Sezgisel |
| Tamimi and Rajan (1997) | | √ | | $\sum w_j T_j$ | | Genetik Algoritma |
| Heady and Zhu (1998) | √ | | | Erkenlik Maliyeti + Gecikme Maliyeti | | Sezgisel |
| Balakrishnan et al. (1999) | | √ | | $\sum w_j E_j + \sum w_j T_j$ | | Karma Tamsayılı Programlama |
| Sivrikaya-Serifoglu and Ulusoy (1999) | | √ | | $w_E \sum E_j + w_T \sum T_j$ | | Genetik Algoritma |

Çizelge 2.2. (devam) Partiye bağlı olmayan, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar.

| Yazarlar | Paralel Makine Yapısı Açısından | | | Amaç Fonksiyonu Açısından | | Yaklaşım |
|----------------------------------|---------------------------------|---|---|---|-------------------------|---|
| | P | Q | R | Teslim Zamanıyla İlişkili | Diğer | |
| Vignier et al. (1999) | √ | | | | Toplam Maliyet | Yinelemeli Sezgisel, GA ve Dal-Sınır Alg. oluşan Karma Yöntem |
| Park et al. (2000) | √ | | | $\sum w_j T_j$ | | YSA ve Sezgisel |
| Radhakrishnan and Ventura (2000) | √ | | | $\sum E_j + \sum T_j$ | | Karma Tamsayılı Programlama ve Tavlama Benzetimi |
| Zhu and Heady (2000) | | | √ | $\sum w_j E_j + \sum w_j T_j$ | | Karma Tamsayılı Programlama |
| Gendreau et al. (2001) | √ | | | | C_{max} | Alt sınırlar, Böl ve Birleştir Sezgiseli |
| Hurink and Knust (2001) | √ | | | | C_{max} | Karmaşıklık sonuçları |
| Kruz and Askin (2001) | √ | | | | C_{max} | Tamsayılı Programlama, GSP, Sezgiseller |
| Weng et al. (2001) | | | √ | | $\sum w_j C_j$ | Sezgiseller |
| Hiraishi et al. (2002) | √ | | | Erken ve Geç biten işlerin ağırlıklı sayısı | | Polinom Algoritma |
| Mendes et al. (2002) | √ | | | | C_{max} | Yasaklı Arama, Memetik Algoritma |
| Fowler et al. (2003) | √ | | | $\sum w_j T_j$ | $C_{max}, \sum w_j C_j$ | Melez Genetik Algoritma |
| Kim et al. (2003) | √ | | | $\sum w_j T_j$ | | Dört-Safhalı Sezgisel (3. safhası Yasaklı Arama) |
| Kim and Shin (2003) | √ | | √ | L_{max} | | Sınırlı Yasaklı Arama |
| Bilge et al. (2004) | | √ | | $\sum T_j$ | | Yasaklı Arama |
| Anglani et al. (2005) | √ | | | | TSC | Karma Tamsayılı Programlama |
| Feng and Lau (2005) | √ | | | $\sum w_j E_j + \sum w_j T_j$ | | Squeaky Wheel Optimization isimli Bir Meta Sezgisel |
| Nessah et al. (2005) | √ | | | | $\sum C_j$ | Yeter ve Gerek Şart Durumu, Sezgisel, Alt Sınır |
| Tahar et al. (2006) | √ | | | | C_{max} | Sezgisel |

Parker et al. (1977), sıra bağımlı geçiş maliyetlerini (sequence dependent changeover costs) içeren kapasite kısıtlı paralel makine problemini bir araç rotalama problemi olarak ele almış ve toplam geçiş maliyetlerini (total changeover costs) en küçükleyecek şekilde algoritmalar tanımlamışlardır. Toplam hazırlık maliyeti bazında Dearing and Henderson (1984), modellerine uygun problemin çözümü için tamsayılı doğrusal programlamayı kullanmışlardır. Sumichrast and Baker (1987), Dearing and Hendersen'nin (1984) çalışmalarının sonuçlarını iyileştiren tamsayılı programlama temelli bir sezgisel yöntem önermişlerdir (Allahverdi et al., 1999).

Ovacık and Uzsoy (1995), hazırlık sürelerinin işlem süreleriyle sınırlandırıldığı yarı iletken ölçüm tesislerindeki (semiconductor testing facilities) $P|ST_{sd}|C_{max}$ ve $P|ST_{sd}|L_{max}$ problemleri üzerinde çalışmışlardır. Liste çizelgelemelerin (list scheduling) baskın olmadıklarını (non-dominant) gösteren bir örnek sunmuşlardır ve liste çizelgeleme algoritmaları için en kötü durum hata sınırlarını (worst-case error bounds) geliştirmişlerdir. Franca et al. (1996), Ovacık and Uzsoy'un (1993) probleminde hazırlık sürelerinin üzerinde bir kısıtlamanın olmadığı durumu C_{max} amacı altında ele almışlar ve yasaklı aramanın da içinde bulunduğu üç-safhalı bir sezgisel geliştirmişlerdir. Franca et al. (1996) tarafından ele alınan problemin dinamik yapıda olanını Ovacık and Uzsoy (1995), en büyük sapmayı en küçükleme amacı altında sezgisel yaklaşımlar (rolling horizon heuristics) sunmuşlardı. Önerdikleri sezgiseller, yerel arama yöntemleriyle ilişkilendirildiğinde bile sevk etme kurallarından daha iyi sonuçlar vermiştir. Guinet and Dussauchoy (1993), $P|ST_{sd}|C_{max}$ problemini çözen bir sezgisel yaklaşım olarak doğrusal atama problemini çözen Macar algoritmasının genişletilmiş durumunu kullanmışlardır (Allahverdi et al., 1999).

Lee and Pinedo (1997), $P|ST_{sd}|WTT$ problemini çözen, bir sevk etme kuralını ve tavlama benzetimini de içeren üç-fazlı bir sezgisel sunmuşlardır. Kural, Lee et al. (1997) tarafından tek makine için kullanılan *ATCS* kuralının paralel makinelere uyarlanmış hâlidir (Allahverdi et al., 1999).

Heady and Zhu (1998), bazı işlerin bazı makineler tarafından işlenemediği $P|ST_{sd}$ problemini ele almışlardır. Erkenlik ve gecikme maliyetlerinin toplamını en küçüklemek için bir sezgisel önermişlerdir. Küçük boyutlu problemler için, önerilen yöntemin performansı tamsayı programlama kullanılarak elde edilen optimal çözümle karşılaştırılmıştır. Vignier et al. (1999), iki tipteki makinelerin mevcut olduğu, hem işlem hem de hazırlık sürelerinin makinelere bağlı olduğu, her bir işin geliş ve teslim zamanına sahip olduğu $P|ST_{sd}$ problemini ele almışlardır. Öncelikli olarak uygun bir çözüm bulmayı ve sonra da atama ve hazırlık sürelerine göre maliyeti en küçüklemeyi amaç olarak edinmişlerdir. Yinelemeli sezgiseli, genetik algoritmayı ve dal-sınır algoritmasını içine alan karma bir yöntem önermişlerdir (Allahverdi et al., 2008).

Radhakrishnan and Ventura (2000), $P|ST_{sd}|\sum E_j + \sum T_j$ probleminin kısıtlı boyutu için bir matematiksel programlama formülasyonu, daha büyük boyutu için ise tamamlanan işlerin ağırlıklı sayısını en büyükleme amacı altında bir tavlama benzetimi algoritması önermişlerdir. Genelde problem NP-zor olmasına rağmen, bazı özel durumlar göz önüne alındığında problemin polinom zamanda çözülebileceğini göstermişlerdir (Allahverdi et al., 2008).

Mendes et al. (2002) ve Gendreau et al. (2001), $P|ST_{sd}|C_{max}$ problemini ele almışlardır. Mendes et al. (2002), biri yasaklı arama diğeri ise memetik (memetic) algoritma tabanlı iki sezgisel yöntem önermişlerdir. Gendreau et al. (2001), alt sınırlar tanımlamış ve bir böl-birleştir (divide and merge) sezgisel yaklaşımıyla problemin çözümüne yanıt aramıştır. Kendi yaklaşımlarını yasaklı aramayı içeren önceki sezgisel yaklaşımlarla karşılaştırmış ve yaklaşımlarının benzer kalitedeki sonuçları daha kısa zamanda elde ettiğini göstermişlerdir. Tahar et al. (2006), aynı $P|ST_{sd}|C_{max}$ problemine iş bölünmesi özelliğini (job splitting property) de katarak incelemişlerdir. Probleme, doğrusal programlama modeline dayalı bir sezgisel yaklaşım önermişlerdir. Önerdikleri yaklaşımın performansını, bir alt sınıra dayalı yöntemin çözümleriyle test etmişlerdir (Allahverdi et al., 2008).

Hurink and Knust (2001), $P|ST_{sd,prec}|C_{max}$ problemini bölümlenme ve sıralama (partitioning and sequencing) olmak üzere iki boyutta ele almışlardır. Kruz and Askin (2001), $P|ST_{sd,r_j}|C_{max}$ problemi için tamsayı programlama formülasyonu sunmuşlardır. Ayrıca genetik algoritma ve çoklu-uyum tabanlı (multi-fit based) yaklaşımları da içeren bir çok sezgisel yöntem geliştirmişler ve deneysel olarak test etmişlerdir. Gezgin satıcı problemini (GSP) sezgisel çözümlerinin bir parçası olarak kullanmışlardır. Şöyle ki; işler makinelerle atandıktan sonra, her bir makinede en iyi sıralamayı bulmak için bir GSP formüle edilip çözülmüştür. GSP’nde asimetrik uzaklıklar, sıra bağımlı hazırlık sürelerine karşılık gelmektedir. Kim and Shin (2003), özdeş ve özdeş olmayan makine durumlarının her ikisi için $P|ST_{sd,r_j}|L_{max}$ problemine bir kısıtlı yasaklı arama (restricted tabu search) algoritması önermişlerdir. Kısıtlı arama algoritmasının, arama gücünü bariz azalttığı görülmüştür (Allahverdi et al., 2008).

Weng et al. (2001), $R|ST_{sd}|\sum w_j C_j$ problemini ele almışlardır. Probleme yedi basit sezgisel yaklaşımlar sunmuşlar ve deneylerle içlerinden bir tanesinin diğerlerinden daha üstün olduğunu göstermişlerdir. Fowler et al. (2003), $P|ST_{sd,r_j}|\sum w_j C_j$, $P|ST_{sd,r_j}|\sum w_j T_j$ ve $P|ST_{sd,r_j}|C_{max}$ problemlerine bir karma genetik algoritma (hybrid genetic algorithm) önermişlerdir. Karma genetik algoritmada genetik algoritma, işleri makinelerle atamada; sevk etme kuralları ise her bir makinedeki işleri çizelgelemede kullanılmıştır. Performans ölçüleri dikkate alındığında önerilen karma genetik algoritmanın önceki algoritmalara göre daha iyi sonuç verdiği gösterilmiştir. Nessah et al. (2005), $P|ST_{sd,r_j}|C_{max}$ problemi incelemişlerdir. Yerel en iyi çözüm için bir gerek ve yeter şart sunmuşlar ve bu şarta dayalı bir sezgisel yaklaşımda bulunmuşlardır. Ayrıca bir alt sınır da geliştirmişlerdir. Önerdikleri sezgiseli, geliştirdikleri alt sınırla kıyaslamak suretiyle rassal ürettikleri problem üzerinde denemişlerdir (Allahverdi et al., 2008).

Tamimi and Rajan (1997), $Q|ST_{sd}|\sum w_j T_j$ problemine bir genetik algoritma önermiştir. Genetik algoritmalarında mutasyon oranını, eşleme oranını ve insertion rate’i dinamik olarak değiştirmişlerdir. Park et al. (2000), $P|ST_{sd}|\sum w_j T_j$ probleminde öncelik kuralını hesaplamadaki parametre değerlerini elde etmek için sinir ağı kullanımını önermiştir. Elde ettikleri sonuçlar, kendi önerdikleri yaklaşımın önceki yaklaşım üzerinde üstünlük kurduğunu göstermiştir. Kim et al. (2003), aynı problem

için üçüncü safhası yasaklı arama olan dört safhalı bir sezgisel sunmuşlardır. Bilge et al. (2004), $P|ST_{sd,r_j}|\sum T_j$ problemine bir yasaklı arama algoritması önermişlerdir. Yasaklı aramanın bir çok anahtar bileşenlerinin alması gerektiği değerler araştırılmış ve en iyi değerleri belirlenmiştir. Kendi sezgisellerini sıfır ağırlıklı erkenliğin olduğu Sivrikaya-Serifoglu and Ulusoy'un (1999) genetik algoritması ile karşılaştırmışlar ve kendi sezgisellerinin Sivrikaya-Serifoglu ve Ulusoy'un (1999) üzerinde üstünlük kurduğunu göstermişlerdir (Allahverdi et al., 2008).

Sivrikaya-Serifoglu and Ulusoy (1999), farklı hızlara sahip iki tipteki makinelerin olduğu $Q|ST_{sd,r_j}|w_E\sum E_j+w_T\sum T_j$ problemini ele almışlardır. Buradaki performans ölçütünün anlamı, erkenlik ve gecikme için ağırlıkların tüm işlerde aynı olduğunu ifade etmektedir. çaprazlama operatörü (crossover operator) olan ve olmayan olmak üzere iki tip genetik algoritma oluşturmuşlardır. Eşleşme operatörüne sahip genetik algoritmanın zor ve büyük boyutlu problemlerde daha iyi sonuç verdiğini gözlemlemişlerdir. Balakrishnan et al. (1999), $\sum w_j E_j + \sum w_j T_j$ amacı altında benzer makineli genel durumu incelemişlerdir. Problemlerine bir karma tamsayılı programlama formülasyonu oluşturmuşlardır. Zhu and Heady (2000), $R|ST_{sd}|\sum w_j E_j + \sum w_j T_j$ problemini ele almışlardır. Dokuz iş ve üç makineli duruma ait en iyi çözümü makul zamanda sağlayan bir karma tamsayılı programlama formülasyonu geliştirmişlerdir (Allahverdi et al., 2008).

Toplam hazırlık maliyetlerinin en küçüklendiği $P|SC_{sd}$ problemi, Anglani et al. (2005) tarafından incelenmiştir. İşlerin işlem sürelerinin belirsiz olduğu bir durumu ele almışlar ve problemlerini çözmek için bir bulanık matematiksel programlama yaklaşımı sunmuşlardır. Ayrıca problemlerinin bir karma tamsayılı doğrusal programlamaya dönüştürülebildiğini göstermişlerdir. Ek olarak; büyük boyutlu problemler için bir yaklaşımsal model önermişler ve yaklaşımsal model çözümünün en iyi çözümden ortalama sapmasının %1.5'ten daha az olduğunu göstermişlerdir (Allahverdi et al., 2008).

2.3.3. Partiye bağılı, sıradan bağımsız hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar

Partiye bağılı, sıradan bağımsız hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar Çizelge 2.3'te verilmiştir. Tamamlanma zamanını içeren problemler ağırlıktadır. Çözüm yaklaşımı olarak daha çok polinom ve sözde-polinom zamanlı algoritmalar ve sezgisel yöntemler kullanılmıştır.

Özdeş paralel makine problemleri için Bruno and Sethi (1978), toplam elde bulundurma maliyetini (total holding cost) en küçüklemede dinamik programlamayı (DP) kullanmışlardır. En büyük tamamlanma zamanını en küçükleme amacı altında Tang and Wittrock (1985), problemin çözümü için iki sezgisel yöntem kullanmıştır.

Monma and Potts (1989), parti hazırlık süreli iki özdeş makine çizelgeleme problemini ele almıştır. Sabit parti sayısına sahip kesintili (preemptive) ve kesintisiz (non-preemptive) durumlar için en büyük tamamlanma zamanı, en büyük sapma, geciken işlerin sayısı ve toplam ağırlıklı tamamlanma zamanını içeren performans ölçütleri dâhilinde sözde polinom algoritmaların var olduğunu saptamışlardır. Ayrıca parti sayısının keyfi olduğu durumda söz edilen problemlerin NP-zor sınıfına girdiğini göstermişlerdir. Toplam tamamlanma zamanı kriterini içeren problem daha sonra Cheng and Chen (1994) tarafından işlem süreleri eşit alındığında bile ikili (binary) NP-zor sınıfında yer aldığı gösterilmiştir. En büyük tamamlanma zamanlı ve kesintili durum için Monma and Potts (1993), önceki çalışmalarını m -makinelik problemler için geliştirmiş ve iki tip sezgisel yöntem ve en kötü durum performans oranları (worst-case performance ratios) sunmuşlardır. Sezgisellerinde Chen (1993) tarafından daha iyi en kötü durum performans oranları elde etmek için değişiklikler yapılmıştır (Allahverdi et al., 1999).

Çizelge 2.3. Partiye bağlı, sıradan bağımsız hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar.

| Yazarlar | Paralel Makine Yapısı | | | Amaç Fonksiyonu Açısından | | Yaklaşım |
|-------------------------------|-----------------------|---|---|---------------------------|-------------------------|-------------------------------------|
| | P | Q | R | Teslim Zamanıyla İlişkili | Diğer | |
| Bruno and Sethi (1978) | √ | | | | Toplam Bulundurma Mali. | Dinamik Programlama |
| Tang and Wittrock (1985) | √ | | | | Cmax | Sezgisel |
| Wittrock (1986) | √ | | | | Cmax | Sezgisel |
| Monma and Potts (1989) | √ | | | Lmax, NLJ | Cmax, WTCT | Sözde Polinom Algoritmalar |
| Cheng and Chen (1994) | √ | | | | TCT | Karmaşıklık Hesaplama Algoritması |
| Monma and Potts (1993) | √ | | | | Cmax | Sezgiseller |
| Chen (1993) | √ | | | | Cmax | Sezgisel |
| So (1990) | √ | | | | Toplam ödül fonk. | Sezgiseller |
| Tang (1990) | √ | | | | Cmax | Sezgiseller, Alt Sınırlar |
| Wittrock (1990) | √ | | | | Cmax | Yinelemeli Aç Gözlü Sezgisel |
| Dietrich (1989) | | | √ | | f[Cmax,TCT] | İki Safhalı Sezgisel |
| Schutten and Leussink (1996) | √ | | | Lmax | | Dal-Sınır Algoritması |
| Brucker et al. (1998) | √ | | | Lmax | | Yaklaşım Algoritması |
| Liaee and Emmons (1997) | √ | | | | $\sum C_j$ | Karmaşıklık Hesapları |
| Liu et al. (1999) | √ | | | | $\sum C_j$ | Sözde Polinom Zamanlı Algoritma |
| Gambosi and Nicosia (2000) | √ | | | | Cmax | Algoritma, alt-sınır, üst-sınır |
| Webster and Azizoglu (2001) | √ | | | | $\sum w_j f_j$ | Dinamik Programlama |
| Yi and Wang (2001a) | √ | | | | $\sum f_j$ | Sezgisel, Yasaklı Arama |
| Yi and Wang (2001b) | √ | | | | $\sum f_j$ | Sezgisel, Alt-Sınır |
| Blazewicz and Kovalyov (2002) | √ | | | | $\sum C_j$ | Polinom Zamanlı Dinamik Programlama |
| Azizoglu and Webster (2003) | √ | | | | $\sum w_j f_j$ | Dal-Sınır Algoritması |
| Chen and Powell (2003) | √ | | | $\sum w_j U_j$ | $\sum w_j C_j$ | Dal-Sınır Algoritması |

Çizelge 2.3 (devam) Partiye bağlı, sıradan bağımsız hazırlık süreli paralel makine çizelgeleme problemlerini içeren çalışmalar.

| Yazarlar | Paralel Makine Yapısı | | | Amaç Fonksiyonu Açısından | | Yaklaşım |
|----------------------------|-----------------------|---|---|-------------------------------|-----------------|--|
| | P | Q | R | Teslim Zamanıyla İlişkili | Diğer | |
| Yi and Wang (2003) | √ | | | $\sum w_j E_j + \sum w_j T_j$ | | Bulanık Mantık içeren Genetik Algoritma |
| Wilson et al. (2004) | √ | | | | C_{max} | Sezgisel, Genetik Algoritma |
| Yi et al. (2004) | √ | | | | $\sum f_j$ | Bulanık Mantık içeren Genetik Algoritma |
| Chen and Wu (2006) | | | √ | $\sum T_j$ | | Sezgisel |
| Dunstall and Wirth (2005a) | √ | | | | $\sum w_j C_j$ | Dal-Sınır Algoritması, Baskın Kurallar |
| Dunstall and Wirth (2005b) | √ | | | | $\sum w_j C_j$ | Sezgiseller |
| Crauwels et al. (2006) | √ | | | | Hazırlık sayısı | Tamsayılı Programlama, Sezgisel |
| Leung et al. (2008) | √ | | | | $\sum C_j$ | Yaklaşım Algoritması |
| Cheng and Kovalyov (2000) | | | √ | Teslim Tarihleri | | Dinamik Programlama |
| Lin and Jeng (2004) | √ | | | $L_{max}, \sum U_j$ | | Sözde Polinom Dinamik Programlama, Sezgiseller |
| Yang (2004) | √ | | | | $\sum C_j$ | Yapılandırıcı Sezgiseller |

So (1990), Tang (1990) ve Wittrock (1990) sırasıyla iş tipleri ve aileleri için minor ve major hazırlıklı özdeş paralel makine çizelgeleme problemini ele almışlardır. So (1990), makine kapasitelerinin sabit olduğu varsayımı altında toplam ödül fonksiyonunu (total reward function) en büyükmek için üç sezgisel yaklaşım geliştirmiştir. Her ne kadar toplam ödülü, performans ölçütü olarak düşünmüş olsa da geliştirilen sezgisellerin en büyük tamamlanma zamanını en küçükleyecek şekilde değiştirilebileceğini de ifade etmiştir. Diğer taraftan Tang (1990), en büyük tamamlanma zamanını en küçükleyen statik problemi çözmek için alt sınırlar ve sezgiseller sunmuştur. Wittrock (1990), iş aileleri ve aile içerisindeki işleri sıralamada sırasıyla major ve minor hazırlıklarda fayda sağlayan yinelemeli aç gözlü sezgisel yaklaşımını (iterative greedy heuristic) geliştirerek Tang'ın (1990) sonuçlarını

iyileştirmiştir. Dietrich (1989), major ve minor hazırlıklı farklı paralel makine problemini ele almış ve çizelge en büyük tamamlanma zamanını (schedule makespan) ve toplam akış süresini içeren doğrusal bir fonksiyonu enküçükleme için iki safhalı bir sezgisel yaklaşımda bulunmuştur (Allahverdi et al., 1999).

Maksimum sapmayı enküçükleme kriteri altında Schutten and Leussink (1996), işlerin geliş zamanının, teslim zamanının ve parti hazırlıklarının olduğu özdeş paralel makine problemi için bir dal sınır algoritması geliştirmişlerdir. İçlerinden birinin kesinti durumunu ele aldığı iki alt sınır tanımlamışlardır. Brucker et al. (1998), bu problemin çözümü için yaklaşım algoritması geliştirmişlerdir (Allahverdi et al., 1999).

Liu et al. (1999), ortak hazırlık zamana sahip $P2|ST_{si,b}, p_j=p|\sum C_j$ çoklu aile problemi (multiple family problem) için sözde-polinom zamanlı bir algoritma sunmuş ve problemin NP-zor karmaşıklığını ispatlamıştır. Liaee and Emmons (1997), tüm ailelerin aynı iş sayısını içermediği grup teknolojisi varsayımı altında aynı problemin tipik (ordinary) NP-zor sınıfına girdiğini kanıtlamışlardır. Blazewicz and Kovalyov (2002), $P|ST_{si,b}|\sum C_j$ probleminin grup teknolojisi varsayımı altında üst düzey NP-zor sınıfına girdiğini kanıtlamış ve bir özel durum için polinom zamanlı dinamik programlama algoritması sunmuştur.

Leung et al. (2008), her bir işin işlem süresinin bekleme süresinin, yani işin ait olduğu partinin işlenmeye başlaması ile işin işlenmeye başlaması arasındaki süre, bir basamak fonksiyonu (step function) olduğu $Pm|ST_{si,b}=s|\sum C_j$ problemini ele almışlardır. Her iş i için; o işe ait bekleme süresi belli bir D eşik değerinden az ise, işlem süresinin $p_i=a_i$; diğer durumlarda, işlem süresinin $p_i=a_i+b_i$ olmasını gerektirir. Bu problemin tek makine ve $b_i=b$ olduğu durumda bile üst düzey NP-zor sınıfına girdiğini kanıtlamışlardır (Allahverdi et al., 2008).

Yi and Wang (2001b), işlerin çizelgenin başlangıç zamanda hazır olduğu varsayımı altında $P|ST_{si,b}|\sum f_j$ problemi için bir alt sınır belirlerken; bir başka çalışmalarında ise Yi ve Wang (2001a) bir yasaklı arama algoritmasını önermişlerdir. Yi et al. (2004), aynı probleme genetik algoritma içerisinde bulanık mantığın gömülü

olduğu bir algoritma da sunmuşlardır. Webster and Azizoglu (2001) ve Azizoglu and Webster (2003), aynı problemi ağırlıklı amaç fonksiyonuyla, yani $P|ST_{si,b}|\sum w_j f_j$ ya da eş değer olarak $P|ST_{si,b}|\sum w_j C_j$, ele almışlardır. Algoritmaların kullanıldığı problemin özelliklerini de belirterek, iki dinamik programlama algoritması (geriye doğru ve ileriye doğru olmak üzere) önermişlerdir. Azizoglu and Webster (2003), problem için bir çok dal sınır algoritmaları tanımlamış ve her bir algoritma performans açısından değerlendirilmiştir. Makine sayısına bağlı olmak suretiyle algoritmaların 15 ila 25 işe kadar en iyi çözümü hızlı bir şekilde ürettiğini göstermişlerdir. Chen and Powell (2003), aynı probleme *dal sınır algoritmaları tabanlı sütun türetme* (column generation based on branch and bound algorithms) önermişler ve 40 iş, 4 makine ve 6 aileye kadarki problemlere en iyi sonuçları elde etmişlerdir. Dunstall and Wirth (2005a), aynı problem için bir başka dal sınır algoritması tanımlamış ve algoritmalarının Azizoglu ve Webster (2003)'in algoritmasından daha iyi sonuç verdiğini göstermişlerdir. Kendi dal sınır algoritmalarını kullanarak 25 iş ve 8 aileye kadarki problemleri çözmüşlerdir. Dunstall ve Wirth (2005b), aynı probleme bir çok basit sezgiseller tanımlamışlardır (Allahverdi et al., 2008).

Chen and Powell (2003), $P|ST_{si,b}|\sum w_j U_j$ problemine *dal sınır algoritmaları tabanlı sütun türetme* önermişlerdir. 40 iş, 6 aile ve 4 makineli problemlere kadar en iyi sonuçları elde etmişlerdir. Chen and Wu (2006), $R|ST_{si,b}|\sum T_j$ problemini ele almıştır ve eşik kabul (threshold-accepting), tabu listesi (tabu list) ve iyileştirme prosedürlerine (improvement procedures) dayanan bir sezgisel yöntem sunmuşlardır. Sayısal analizleri, sezgiselleriyle elde ettikleri sonuçların tavlama benzetimiyle elde edilenlerden açık bir şekilde üstün olduğunu göstermiştir. Yi and Wang (2003), işlerin teslim zamanlarının ortak olduğu $P|ST_{si,b}|\sum w_j E_j + \sum w_j T_j$ problemini ele almışlardır. Problemi çözmek için genetik algoritması içine entegre edilmiş bir bulanık mantık önermişlerdir (Allahverdi et al., 2008).

Gambosi and Nicosia (2000), $P|ST_{si,b}|C_{max}$ problemine bir on-line algoritma önermişler ve rekabetçi oranına (competitive ratio) bir üst sınır türetmişlerdir. Ayrıca herhangi bir on-line algoritma için bir de alt sınır türetmişlerdir. Crauwels et al. (2006),

bir çok performans ölçütünü kapsayan $P|ST_{si,b},r_j,d_j$ problemine bir tamsayılı programlama formülasyonu ve sezgisel yöntemler sunmuştur (Allahverdi et al., 2008).

Cheng and Kovalyav (2000), *parti elverişlilik modeli* (batch availability) altında tek aileli, işlerin son teslim zamanlarının olduğu, farklı paralel makineleri içeren problem üzerine çalışmışlardır. Bir dinamik programlama algoritması önermişlerdir.

Lin and Jeng (2004), *parti elverişlilik modeli* altında $P|ST_{si,b}|L_{max}$ ve $P|ST_{si,b}|\sum U_j$ tek aile problemlerini ele almışlardır. Sabit makine sayısı için sözde polinom zamanlı bir dinamik programlama algoritması ile en küçük tamamlanma zamanı ve en küçük sapma öncelik kurallarına sahip sezgiseller sunmuşlardır.

Wilson et al. (2004), döşeme mobilya üretiminde parti hazırlık sürelerinin ortak olduğu durum için $P|ST_{si,b},r_j|C_{max}$ problemi üzerine çalışmışlardır. Parti bölünmeli ve çizelgelemeli bir sezgisel (batch splitting and scheduling heuristic) önermişler ve bu sezgiseli bir genetik algoritma içinde ele almışlardır.

Gerodimos et al. (2000) tarafından ele alınan tek makineli çizelgeleme problemine benzer şekilde Yang (2004), her bir işin standart ve spesifik olmak üzere iki bileşenden meydana geldiği düşüncesiyle bir paralel makine çizelgeleme problemini ele almıştır. Toplam tamamlanma zamanının en küçüklenmesi amacı altında iki yapılandırıcı sezgisel yöntem (constructive heuristics) önermiştir (Allahverdi et al., 2008).

2.3.4. Partiye bağlı, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar

Partiye bağlı, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar Çizelge 2.4'te verilmiştir. Özdeş paralel makineli

sistemler kadar farklı paralel makineli sistemler de ele alınmıştır. Bu alanda yer alan çok amaçlı bir problem için de sezgisel bir yaklaşımda bulunulmuştur.

Çizelge 2.4. Partiye bağlı, sıraya bağımlı hazırlık süreli Paralel Makine Çizelgeleme problemlerini içeren çalışmalar.

| Yazarlar | Paralel Makine Yapısı | | | Amaç Fonksiyonu Açısından | | Yaklaşım |
|---------------------------|-----------------------|---|---|---|----------------|--|
| | P | Q | R | Teslim Zamanıyla İlişkili | Diğer | |
| Bitran and Gilbert (1990) | √ | | | | TSC | Sezgisel |
| Ghosh (1994) | √ | | | | TCT, WTCT | Dinamik Programlama |
| Akkiraju et al. (2001) | | | √ | $\sum w_j T_j$ ve $\sum w_j E_j$ ve TST | | Eş Zamanlı Olmayan Takım Mimarisi |
| Jeong et al. (2001) | | | √ | | $\sum f_j$ | Yapılandırıcı Sezgiseller |
| Eom et al. (2002) | √ | | | $\sum w_j T_j$ | | Yasaklı Aramanın yer aldığı 3 Safhalı Sezgisel |
| Kim et al. (2002) | | | √ | $\sum T_j$ | | Tavlama Benzetimi |
| Chen and Powell (2003) | √ | | | $\sum w_j U_j$ | $\sum w_j C_j$ | Dal-Sınır Algoritması |
| Kim et al. (2003) | | | √ | $\sum w_j T_j$ | | Yapılandırıcı Sezgiseller ve Tavlama Benzetimi |
| Yalaoui and Chu (2003) | √ | | | | C_{max} | Sezgisel |
| Dupuy et al. (2005) | √ | | | $\sum T_j$ | | Tavlama Benzetimi |
| Arnaout et al. (2006) | | | √ | | $\sum w_j C_j$ | Sezgiseller |

Bitran and Gilbert (1990), hazırlık maliyetlerinin biri diğerinden daha büyük olmak üzere iki büyüklüğe sahip olduğu $P|SC_{sd,b}|TSC$ problemini çözmek için sezgisel yöntemler geliştirmişlerdir. Ele aldıkları kriterin en küçüklenmesinin kabaca büyük olanının sayısının en küçüklenmesine eşit olduğunu göstermişlerdir. Ghosh (1994), parti hazırlık sürelerinin üçgen eşitsizliğine uyduğu $P|ST_{sd,b}|TCT$ ve $P|ST_{sd,b}|WTCT$ problemlerini ele almıştır. Parti veya makine sayısının keyfi olduğu durum için, ağırlıklı tamamlanma zamanı dikkate alındığında problemin üst düzey NP-zor sınıfına girdiğini

göstermişlerdir. Parti sayılarının keyfi, işlerin aynı olduğu durumda bile problem yine üst düzey NP-zor sınıfına girmektedir. Oysa ki; sabit bir parti sayısı düşünüldüğünde, toplam tamamlanma zamanı problemi polinom zamanda çözülmektedir (Allahverdi et al., 1999).

Kim et al. (2002), aynı aile içindeki işlerin teslim zamanlarının aynı olduğu $R|ST_{sd,b}|\sum T_j$ problemini ele almışlar ve probleme bir tavlama benzetimi yaklaşımı önermişlerdir. Sayısal analizler neticesinde tavlama benzetiminin bir komşuluk arama yönteminden daha iyi sonuçlar verdiğini göstermişlerdir. Eom et al. (2002), $P|ST_{sd,b}|\sum w_j T_j$ problemi için bir üç-safhalı sezgisel yaklaşımda bulunmuşlardır. Algoritmanın son ayağında yasaklı aramayı kullanmışlardır. Dupuy et al. (2005), takvim kısıtları (calendar constraints) olarak adlandırılan kısıtlarlar birlikte $P|ST_{sd,b,r_j}|\sum T_j$ problemini ele almışlardır. Bir çok komşuluk mekanizmalarıyla birlikte bir tavlama benzetimi yaklaşımında bulunmuşlardır. Sayısal deneyler neticesinde; kendi önerdikleri komşuluk mekanizmalarının, bir çok açgözlü sezgiseller ve basit tavlama benzetimi prosedürüyle karşılaştırıldığında kısa sürede daha iyi sonuçlar elde ettiğini göstermişlerdir (Allahverdi et al., 2008).

Kim et al. (2003), $R|ST_{sd,b}|\sum w_j T_j$ problemi üzerine çalışmışlardır. Problemden makineler, özdeş makine grupları olarak sınıflandırılmıştır. Her bir iş, aynı operasyon sayısına sahiptir, öyle ki; eş zamanda farklı makinelerde işlenebilmektedirler. Bir iş, son operasyonu bittiği vakit tamamlanmaktadır. İşlem süreleri, işe ve makine grubuna bağlıdır. İş ağırlıkları, iş teslim zamanlarıyla ters orantılı olarak değişmektedir. Sıra bağımlı hazırlık süresi, farklı işlerin operasyonlarının partileri arasında meydana gelmektedir. Yazarlar, yarı iletken üretimden gerçek problemler ele alarak, en erken ağırlıklı teslim zamanı ve en kısa ağırlıklı işlem süresi gibi sıralama kuralları, partileme sezgiseli (batching heuristic) ve tavlama benzetimi gibi bir çok yapılandırıcı sezgiseller sunmuş ve bunları sayısal olarak test etmişlerdir.

Yalaoui and Chu (2003), bir işin birçok parçaya bölünebildiği ve bu parçalarında paralel olarak işlenebildiği $P|ST_{sd,b}|C_{max}$ problemi için sezgisel bir yöntem önermişlerdir.

Sezgiselde gezgin satıcı problemine indirgeme yöntemi kullanılmıştır (Allahverdi et al., 2008).

Chen and Powell (2003), $P|ST_{sd,b}|\sum w_j C_j$ ve $P|ST_{sd,b}|\sum w_j U_j$ problemleri için *dal sınır algoritmaları tabanlı sütun türetme* önermişlerdir. Sayısal analizleri algoritmaların orta büyüklükteki problemleri, yani 40 iş, 4 makine ve 6 aileye kadarki problemleri, en iyi olarak çözdüğünü göstermişlerdir. Arnaout et al. (2006), işlem süreleri ve hazırlık sürelerinin stokastik olduğu $R|ST_{sd,b}|\sum w_j C_j$ problemine birçok sezgisel yöntemler önermişlerdir.

Akkiraju et al. (2001), kağıt üretiminde $\sum w_j T_j$, $\sum w_j E_j$ ve TST gibi bir çok amacı ele alarak $R|ST_{sd,b}$ problemini incelemişlerdir. *Eş Zamanlı Olmayan Takım Mimarisi* (Asynchronous Team architecture) olarak isimlendirilen bir sezgisel yöntem önermişlerdir. Başlangıç çözümler, ilk önce uzmanlar ve bilgisayar programları tarafından üretilmiştir. Sonrasında bu çözümler iyileştirilmiştir. Nihayetinde de en iyi sonuçları içeren Pareto kümeleri karar vericiye sunulmuştur.

Jeong et al. (2001), TFT LCD (Thin Film Transistor Liquid Crystal Display) montaj sürecinde gözlenen $R|ST_{sd,b}|\sum f_j$ problemi üzerinde çalışmışlardır. Amaç fonksiyonları, ortalama akış süresi ve ürün talebinden sapmanın bir doğrusal birleşiminden oluşmaktadır. İki belirgin yapılandırıcı sezgisel algoritma geliştirilmiştir (Allahverdi et al., 2008).

BÖLÜM 3

İŞLERİN BÖLÜNEBİLİR OLDUĞU, SIRADAN BAĞIMSIZ HAZIRLIK SÜRELERİNİ İÇEREN ÖZDEŞ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ

3.1 Çizelgelemenin Tanımı Ve Çizelgeleme Problemi

Çizelgeleme, istenilen sürede istenilen çıktıların elde edilmesi için yerine getirilmesi gereken tüm eylemlerin kaynaklar ile eylemler arasındaki ilişkisel kısıtların ve zaman kısıtlarının sağlanarak seçilmesi, örgütlenmesi ve kaynak kullanımlarının zamanlanması sürecidir (Sipper and Bulfin, 1997).

Sabuncuoğlu (1998) çizelgelemeyi, sistem kaynaklarının çeşitli işlere, görevlere veya faaliyetlere zaman temelinde tahsis edilmesi olarak tanımlamıştır. Tersine (1985) tarafından faaliyetlerin nerede ve ne zaman gerçekleştirileceğine karar verme süreci olan çizelgeleme, daha çok girdilerle çıktılar arasındaki zamanlamayla ilgilenir.

Bir başka tanıma göre ise çizelgeleme, zaman ekseninde yerine getirilmesi gereken eylemlere kısıtlı kaynakların tahsisi ile ilgilenen ve bir veya birden fazla amacı eniyilemeye çalışan karar verme sürecidir (Pinedo, 2002: Alpay'dan, 2003).

Kaynaklar ve eylemler çok değişik şekillerde olabilir. Kaynaklar, bir atölyedeki makineler, bir hava alanındaki kalkış pistleri, bir inşaat sahasındaki çalışanlar, bir bilgisayarın işlemcisi vs. olabilir. Eylemler ise, üretim sürecinde yapılması gereken işlemler, bir hava alanındaki kalkış ve inişler, bir yapı projesinin aşamaları, bir bilgisayar programının çalıştırılması vs. olabilir. Her bir eylemin farklı öncelik seviyesi, olası erken başlama zamanı ve bir teslim zamanı olabilir. Amaçlar da farklı şekillerde oluşturulabilir. Bir amaç, en son eylemin tamamlanma zamanının

enküçüklenmesi iken; bir diğeri ise teslim zamanlarından sonra tamamlanan eylemlerin sayısının enküçüklenmesi olabilir (Pinedo, 2002).

Bir üretim sisteminde, atölye içinde çok sayıdaki yarı mamul yığınları veya bir kısım tezgahların çalışırken diğelerinin boş durması gibi durumların gözlemlenmesi çizelgeleme problemlerinin varlığını ortaya koymaktadır. Ayrıca, üretim kayıtları incelendiğinde görülebilecek yüksek seviyede fazla mesai, gecikmiş işlerin varlığı, düşük tezgah ya da işgücü kullanım oranları gibi istatistikler de çizelgeleme problemlerinin belirtileridir (Özkazanç, 1999).

Çizelgeleme problemleri, yerine getirilmesi gereken bir grup görev ve bu görevlerin gerçekleştirilmesinde kullanılacak uygun kaynakları içerirler. Bu iki temel girdilerin nitelikleri iyi belirlenmeli ve alacakları değerler mümkün olduğunca kesin ve doğru olarak hesaplanmalıdır. Ancak bu şekilde doğru zaman planları ortaya çıkarılabilir. Kaynaklar belirlendiğinde çizelgeleme problemlerinin sınırları etkin bir şekilde çizilmiş olmaktadır. Bununla birlikte her görev; kaynak gereksinimi, süresi, başlama ve bitiş zamanları gibi bilgiler cinsinden açıklanır. Aynı zamanda, bu görevler bütünü arasında herhangi bir teknolojik kısıt varsa belirtilir (Özkazanç, 1999).

Baker (1994), çizelgeleme problemlerine çözüm bulmanın oldukça karmaşık bir süreç olduğunu belirtmiştir. Sürecin karmaşıklığı ve ihtiyaç duyulan zaman dolayısıyla, biçimsel yaklaşımlarla çözüm aramanın problem çözümünde etkinliği arttırdığını vurgulamıştır. Biçimsel modellerin, problemlerin anlaşılmasına ve iyi bir çözüm bulunmasına yardımcı olduğuna değinmiştir.

Çizelgeleme konusunda yapılan çalışmalarda çizelgelemenin görüntülenmesi için kullanılan Gantt Şemaları önem arz etmektedir. Gantt şeması basitçe, her bir makine üzerinde işlere ait işlemleri zaman temelinde görüntülemektedir (Schonberger, 1988). Gantt şeması, çizelgenin, makine boş zamanları, toplam üretim zamanı ve iş bekleme süreleri açısından değerlendirilmesine olanak sağlamaktadır. Gantt şeması iyi bir görüntüleme aracı olmasına rağmen, çizelgenin iyileştirilmesine herhangi bir yapısal

yaklaşım getirmemektedir. Analizci, iyileştirilmiş bir çizelge bulabilmek adına kendi sezgilerini kullanmak zorunda kalmaktadır (Johnson, 1954).

Silver et al. (1998), sezgisel yöntemlerin önemini şöyle bir örnekle açıklamaktadır: 4 işin işlem göreceği tek makineli bir sistemi ele alalım. Makinenin işler için ihtiyaç duyduğu hazırlık süresinin işlem sürelerine dâhil edildiğini ve bu hazırlık sürelerinin işlerin sırasına göre değişmediğini, sıradan bağımsız olduğunu, farz edelim. Ayrıca, bir iş bir makinede işlem görmeye başladığında, ilgili makinedeki işlemi bitene kadar herhangi bir kesintiye uğramayacağını (non-preemptive) varsayalım. Bu dört iş için hazırlanacak çizelgede hangisinin ilk, hangisinin ikinci, üçüncü ve dördüncü olarak işlem göreceğine karar vermek gerekmektedir. İlk olarak, işlem görmek için 4 alternatif iş mevcuttur. İşlerden bir tanesi seçildikten sonra ikinci için, 3; üçüncü için, 2 ve son olarak dördüncü iş için 1 alternatifin olacağı açıktır. Diğer bir deyişle, $4 \times 3 \times 2 \times 1 = 4! = 24$ farklı çizelge söz konusudur. Şayet çizelge, bu örnekte olduğu gibi belirli sayıdaki işlerin permütasyonu olarak tanımlanabiliyorsa, bu tip çizelgeler *permütasyon çizelgeler* olarak adlandırılmaktadır. Şimdi işlerin sayısının arttığında ne olacağına bakalım. Şayet 6 iş olsaydı, $6! = 720$; 20 iş olsaydı $20! = 2.43 \times 10^{18}$ alternatif sıralama olacaktı. İşte üstel olarak artan bu alternatif çözümler, çözüm süresinin artmasına neden olmaktadır. Ayrıca her makine için bu sıralamanın yapılacağı düşünülürse, n adet iş ve m adet makinenin mevcut olduğu bir ortamda $(n!)^m$ adet seçenek oluşacaktır. Sadece kombinatoryal çizelgeleme problemleri değil, çok farklı türdeki çizelgeleme problemleri için de benzer durumlar söz konusudur. Silver et al. (1998) tarafından da vurgulandığı gibi çizelgeleme teorisi, genellikle bu tip problemlere eniyi ya da eniyiye yakın çözümler aramaktadır (Seçme, 2006).

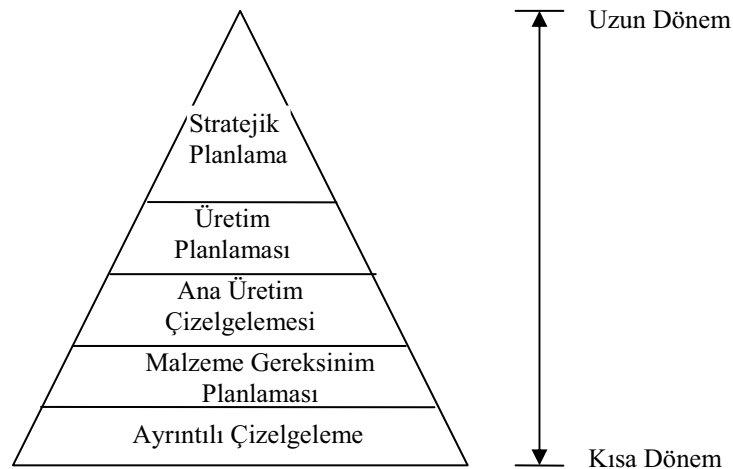
3.2 Bir Örgüt İçerisinde Çizelgelemenin Yeri ve Önemi

Çizelgeleme hem imalat hem de hizmet sektöründe önemli bir yere sahiptir. Günümüz rekabetçi dünyasında pazarda varlığın devamlılığı için etkin çizelgeleme yapmak büyük önem taşımaktadır. Örgütler müşteri isteklerine onlara bildirdikleri tarihlerde cevap vermek durumundadırlar. Aksi hâlde, bir iyi niyet kaybı ortaya

çıkabilecektir. Mevcut kaynakların etkin kullanımını sağlayacak şekilde çizelgeleme yapmak örgütler için ayrıca bir önem taşımaktadır (Pinedo, 1995).

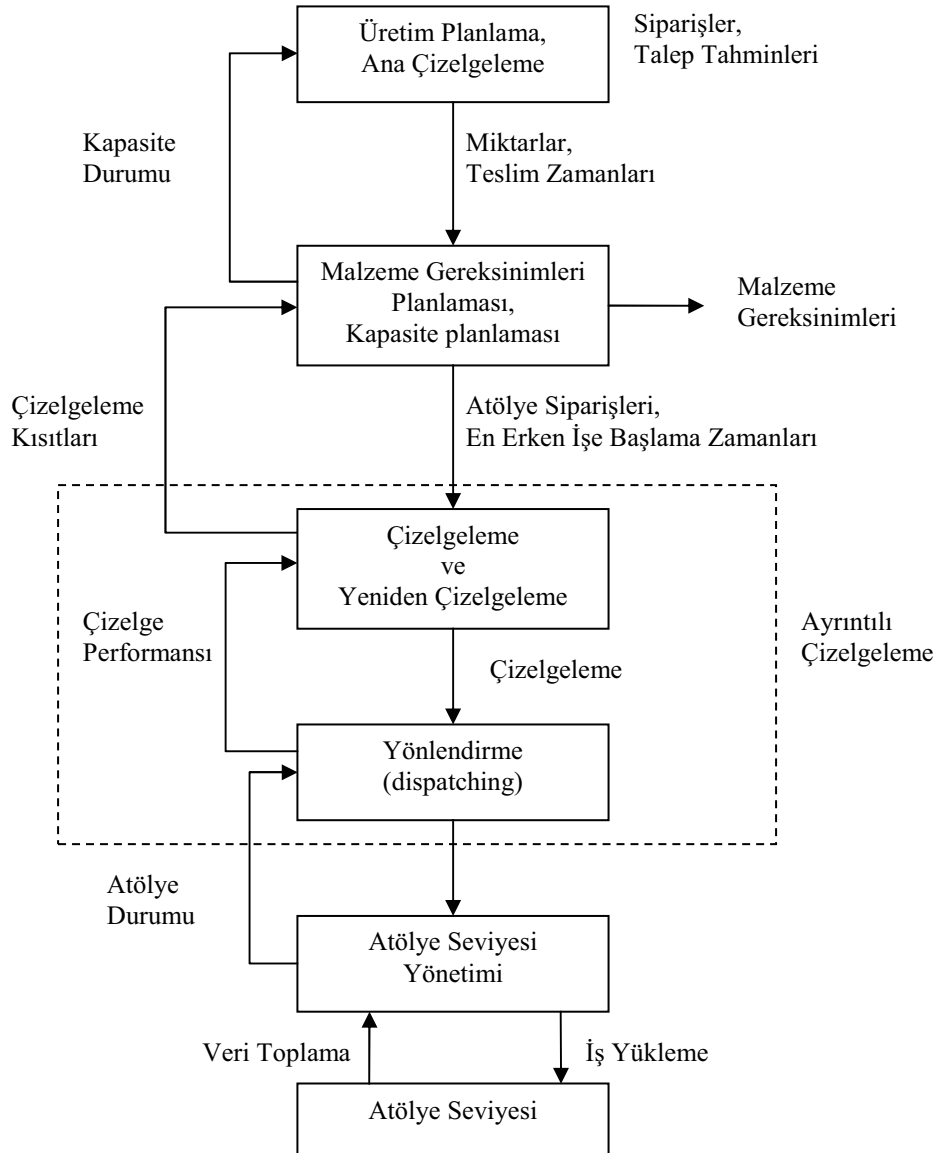
Cheung'a göre; çizelgelemeye olan ihtiyaç, parçaların ve alt parçaların satın alınmasının çizelgelenmesinden, yapım ve montaj sürecinde işlerin ve makinelerin çizelgelenmesinden, paketlerin ve sevkiyatın çizelgelenmesine kadar imalat sürecinin her seviyesinde kendini göstermektedir (Alpay, 2003).

Çizelgeleme işlemi örgüt içerisinde diğer birçok önemli işlev ile etkileşim oluşturmak zorundadır (Pinedo, 1995). Örgütler için iki genel çizelgeleme türü mevcuttur. Bunlar çoğunlukla üç aydan bir yıla kadar değişen bir planlama dönemini kapsayan ve tüm örgüt genelinde üretim seviyesinin planlamasını içeren ana üretim çizelgelemesi ile atölye düzeyi işlerin çizelgelemesi, makinelerin yüklenmesi gibi faaliyetleri içeren ayrıntılı çizelgelemedir (Tersine, 1985). Ayrıntılı çizelgeleme seviyesinde amaç, genellikle gelecek birkaç hafta için yeni gelen işlerin, önceliklerin ve teslim zamanlarının mümkün olduğunca tutarlı bir şekilde ana üretim çizelgesi dâhilinde oluşturulmasını sağlamaktır (Morton and Pentico, 1993). Vollmann et al. (1997), bu iki çizelgeleme faaliyetinin üretim planlama sistemi içerisindeki yerini Şekil 3.1'deki gibi göstermiştir.



Şekil 3.1 Yukarıdan aşağı planlama.

Çalışmada ele alınan çizelgeleme türü olan Ayrıntılı Çizelgelemenin diğer önemli planlama faaliyetleri ile olan etkileşimi Şekil 3.2’de gösterilmiştir.



Şekil 3.2 Bir imalat sistemindeki bilgi akış diyagramı.

3.3 Problemin Tanımlanması

Çalışmada; toplam gecikmeyi enküçükleyen, işlerin bölünebilir özellikte olduğu, n sayıdaki işin m sayıdaki özdeş paralel makinelere atandığı çizelgeleme problemi ele alınmıştır. Paralel makineli bazı üretim sistemlerinde her bir iş, belirli bir teslim zamanını, miktarını ve ürün tipini içeren bir üretim siparişi olarak tanımlanabilmektedir. Bu tarz sistemlerdeki çizelgeleme problemlerinde işler, alt işlere bölünebilmekte ve bu alt işler de aynı anda birden fazla paralel makinede birbirinden bağımsız olarak işlenebilmektedir (Shim and Kim, 2008). Bu özelliği içeren problemler Xing and Zhang (2000) tarafından işlerin bölünebilir olduğu paralel makine çizelgeleme problemi (parallel machine scheduling problem with splitting jobs) olarak adlandırılmaktadır.

Kim et al.'a (2004) göre, baskı devre kartlarının (printed circuit boards, PCB) üretildiği gerçek üretim sistemindeki çizelgeleme problemi bu şekilde ele alınabilmektedir. PCB üretim sisteminde her bir iş, ürün tipi, üretim miktarı ve teslim zamanı tanımlı olan bir PCB ürününe ait sipariş olarak tanımlanabilmektedir. Bu yüzden, siparişi tamamlamak için işlenecek birçok aynı özellikte ürün (identical products) mevcuttur. PCB üretim sistemindeki delme istasyonunda her bir sipariş için bir panel kümesi delinmekte ve ayrıca, paneller üzerindeki delikleri delmek için birçok paralel makine kullanılmaktadır. Bu istasyonda, her bir sipariş (bir panel kümesi için) bir delme makinesinde birlikte delinen küçük sayıda panel gruplarına bölünebilmektedir. İlgili grup birim iş olarak tanımlanabilmekte ve bu panel grupları aynı partiden ya da sipariştten gelseler dahi aynı anda farklı makinelerde işlenebilmektedir. Bir makinede ardı sıra işlenecek aynı özellikteki panel grupları ise alt iş olarak düşünülebilir. Farklı ürünler için delik sayısı ve yerleri farklılık gösterdiğinden, yeni bir ürün makinede işleme tabi tutulmadan önce, o makineye bir NC programının yüklenmesini gerektirmektedir. Diğer bir deyişle, bir makinede işlenecek alt işin ürün tipi, işlenen alt işin ürün tipinden farklılık gösterdiğinde hazırlık süresi uygulanmaktadır.

Problemimiz, Lawler et al. (1993) üç alanlı sınıflandırmasına göre $P|ST_{si}, split|\sum T_j$ şeklinde gösterilebilir.

3.4 Varsayımlar

Ele alınan problemimizde bir işin, kesikli sayıda birim işlere sahip alt işlere bölünebildiği ve bu alt işlerin de paralel makinelerde birbirinden bağımsızca işlenebildiği varsayılmaktadır. Bir işe ait birim işler, o işin işlenebilecek en küçük birimleri olarak tarif edilmekte olup, işlem süreleri ve teslim zamanları bakımından özdeş olarak kabul edilmektedir. Alt iş ise, bir makinede art arda işlenen bir işe ait birim işlerin yani özdeş birim işlerin kümesi olarak tanımlanmaktadır. Tanımlamalardan hareketle; bir işin tüm birim işlerinin teslim zamanlarının yanında, alt işlerinin teslim zamanlarının da o işe ait teslim zamanına eşit olacağı görülmektedir.

Bir başka varsayım da hazırlık süreleri üzerinde yapılmaktadır. Çalışmada, hazırlık sürelerinin iş sıralamasından bağımsız olduğu kabul edilmektedir. Hazırlık faaliyeti yalnızca, bir makinede işlenecek alt iş ile işlenen alt işin aynı işe ait olmadığı durumda uygulanmaktadır.

Çalışmada yer alan diğer varsayımları izleyen şekilde sıralayabiliriz:

- i. Tüm işler, çizelgelemenin başlangıcında hazır durumdadır. İşlerin geliş zamanları sıfırdır ($r_j = 0$).
- ii. İşler birbirinden bağımsız olup, her birinin yalnızca tek bir operasyonu vardır.
- iii. Her bir makine, bir seferde yalnızca bir alt işi işleyebilir.
- iv. Her bir alt iş, yalnızca bir makinede işlenebilir.
- v. Makineler her zaman elverişli konumdadır.

3.5 Terminoloji

Ele alınan çizelgeleme problemine ilişkin terimler, Shim and Kim (2008) tarafından şu şekilde tanımlanmaktadır:

- *İş (job)*, sipariş miktarına ve teslim zamanına sahip bir ürün tipi için üretim sipariş birimi;
- *Birim iş (unit-job)*, bir işin en küçük işlem parçası olup, bir işe ait birim işler birbirinin aynısıdır;
- *Alt iş (sub-job)*, bir makinede ardı sıra işlenen aynı birim işlerin kümesi;
- *İş bölünmesi (job splitting)*; herhangi bir işin bölünebildiği ve ona ait alt parçaların birbirlerinden bağımsızca farklı makinelerde işlenebildiği durum;
- *Sıradan bağımsız hazırlık süresi (sequence-independent setup time)*, hazırlık süresinin yalnızca işlenecek işe bağlı olduğu durumdur.

3.6 Probleme İlişkin Matematiksel Model

Modelde kullanılan gösterimler şu şekildedir:

| | |
|-------|--|
| n | işlerin sayısı |
| m | paralel makine sayısı |
| j | iş göstergesi |
| i | makine göstergesi |
| u_j | j işinin birim iş sayısı (Problemimizde $u_j = u, \forall_j$ için) |
| p_j | j işinin her bir birim işinin işlem süresi |
| s_j | j işinin hazırlık süresi |
| d_j | j işinin teslim zamanı |
| M | En az tüm işlerin işlem sürelerinin ve hazırlık sürelerinin toplamı kadar olan büyük pozitif bir sayı. |

Karar deęişkenlerimiz;

$$x_{ijk} = \begin{cases} 1, & j \text{ işine ait alt iş, } i \text{ makinesinin } k. \text{ pozisyonuna atanmışsa;} \\ 0, & \text{d.d.} \end{cases}$$

$y_{ijk} = i$ makinesinin k . pozisyonuna atanan j işine ait alt işin birim iş sayısı;

$T_{ijk} = i$ makinesinin k . pozisyonundaki j işine ait alt iş için geciktirilebilir süre

şeklinde ele alınsın. Bu durumda modelimiz:

$$\sum_{j=1}^n x_{ijk} \leq 1 \quad k = 1, \dots, n; \quad i = 1, \dots, m \quad (2.1)$$

$$\sum_{k=1}^n x_{ijk} \leq 1 \quad j = 1, \dots, n; \quad i = 1, \dots, m \quad (2.2)$$

$$x_{ijk} \leq y_{ijk} \leq u_j x_{ijk} \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad i = 1, \dots, m \quad (2.3)$$

$$\sum_{i=1}^m \sum_{k=1}^n y_{ijk} = u_j \quad j = 1, \dots, n \quad (2.4)$$

$$T_{ijk} \geq \sum_{s=1}^k \sum_{l=1}^n (p_l y_{ils} + s_l x_{ils}) - d_j - M(1 - x_{ijk})$$

$$j = 1, \dots, n \quad k = 1, \dots, n; \quad i = 1, \dots, m \quad (2.5)$$

$$T_{ijk} \geq 0 \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad i = 1, \dots, m \quad (2.6)$$

$$x_{ijk} \in \{0,1\} \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad i = 1, \dots, m \quad (2.7)$$

$$y_{ijk} \geq 0, \text{ tamsayı} \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad i = 1, \dots, m \quad (2.8)$$

k.a.

$$\text{enküçük} \sum_{j=1}^n \left\{ \text{enbüyük } T_{ijk} \right\} \quad (2.9)$$

$$\begin{matrix} 1 \leq i \leq m \\ 1 \leq k \leq n \end{matrix}$$

şeklinde oluşturulabilir.

(2.1) kısıtı herhangi bir makinenin herhangi bir pozisyonunda en fazla bir işe ait alt iş çizelgelenmesini; (2.2) kısıtı ise bir işe ait alt işin, mevcut makinelerin herhangi birine ait pozisyonlardan en fazla birinde yer almasını sağlamaktadır. Bu kısıt, Shim and Kim'in (2008) çalışmalarındaki 1. önerme ve bunun sonucuyla ilgili olarak oluşturulmuştur. Shim and Kim'in (2008) 1. önermesine göre; aynı işe ait iki ya da daha fazla alt işin aynı makinede olduğu bir çizelgeleme, bu alt işlerin tek bir alt iş olarak birleştirildiği başka bir çizelge tarafından domine edilmektedir. İlgili önermeden hareketle şu sonuca varmışlardır: Her bir işe ait alt işlerin sayısının, makine sayısından fazla olmadığı bir en iyi çözüm mevcuttur. (2.3) kısıtı, karar değişkenlerimiz arasında bağlantı kurmaktadır. Herhangi bir işe ait alt iş, herhangi bir makinenin herhangi bir pozisyonuna atanmışsa o alt işin birim iş sayısının en az bir, en fazla da ilgili işin toplam birim iş sayısı kadar olmasını sağlamaktadır. (2.4) kısıtı, herhangi bir işin bütün makine ve pozisyonlardaki birim işleri toplamının o işe ait birim iş sayısı kadar olduğunu göstermektedir. Aynı zamanda bu kısıt sayesinde karar değişkenlerimizin değer alması dolaylı yoldan sağlanmış olmaktadır. (2.5) kısıtı, herhangi bir işe ait alt işin herhangi bir makine ve pozisyonundaki gecikmesini belirtmektedir. Bu gecikme, eşitsizliğin sağ tarafındaki ilk ifadeyle gösterilen ilgili işe ait alt işin tamamlanma zamanı ile ikinci ifadede yer alan teslim zamanı arasındaki fark kadardır. Üçüncü ifade ise karar değişkenimiz olan x_{ijk} 'nin aldığı değerlere göre bu kısıtı gerekli ya da gereksiz kılmaktadır. Şayet j işine ait bir alt iş, i makinesinin k . pozisyonuna atanmamışsa ($x_{ijk} = 0$) bu ifade $-M$ değerini vermektedir. M pozitif ve çok büyük bir değer olduğundan dolayı ilgili kısıt, (2.6) uyarınca gereksiz bir kısıt hâline gelmektedir. Aksi durumda ise ($x_{ijk} = 1$), bu ifade sıfır değerini vermekte; böylelikle sol taraf ilgili gecikmeyi belirtir hâle gelmektedir. (2.6), (2.7) ve (2.8) ise karar değişkenlerimizin alacağı değerleri ifade etmektedir. Son olarak; (2.9)'da yer alan amaç fonksiyonumuzda, işlerin toplam gecikmesinin en küçüklenmesi yer almaktadır. Bir işe ait alt işlerin teslim zamanları birbirinin aynısı olduğundan, içlerinden en büyük

tamamlanma zamanı da işin tamamlanma zamanını göstereceğinden dolayı, bir işin gecikmesi, o işe ait alt işlerin gecikmeleri arasından en büyük olanına eşit olacaktır. Bu ifade, amaç fonksiyonumuzda parantez içerisindeki ifadeyle gösterilmektedir.

(2.9)'da yer alan amaç fonksiyonumuz doğrusal olmadığından dolayı yeni bir değişken, G_j , tanımlanarak doğrusal hâle getirilebilir. G_j , j işinin gecikme miktarını tanımlasın. Bu durumda modelimiz en son şu şekilde olacaktır:

$$\sum_{j=1}^n x_{ijk} \leq 1 \quad k = 1, \dots, n; \quad i = 1, \dots, m \quad (2.10)$$

$$\sum_{k=1}^n x_{ijk} \leq 1 \quad j = 1, \dots, n; \quad i = 1, \dots, m \quad (2.11)$$

$$x_{ijk} \leq y_{ijk} \leq u_j x_{ijk} \quad i = 1, \dots, m \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad (2.12)$$

$$\sum_{i=1}^m \sum_{k=1}^n y_{ijk} = u_j \quad j = 1, \dots, n \quad (2.13)$$

$$T_{ijk} \geq \sum_{s=1}^k \sum_{l=1}^n (p_l y_{ils} + s_l x_{ils}) - d_j - M(1 - x_{ijk})$$

$$i = 1, \dots, m \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad (2.14)$$

$$T_{ijk} \geq 0 \quad i = 1, \dots, m \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad (2.15)$$

$$G_j \geq T_{ijk} \quad i = 1, \dots, m \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad (2.16)$$

$$x_{ijk} \in \{0,1\} \quad i = 1, \dots, m \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad (2.17)$$

$$y_{ijk} \geq 0, \text{ tam sayı} \quad i = 1, \dots, m \quad j = 1, \dots, n \quad k = 1, \dots, n; \quad (2.18)$$

$$G_j \geq 0 \quad j = 1, \dots, n \quad (2.19)$$

k.a.

$$\text{enküçük} \sum_{j=1}^n G_j \quad (2.20)$$

3.7 Problemin Zorluğu

Toplam gecikmeyi enküçükleyen, işlerin bölünebilir özellikte olduğu özdeş paralel makine çizelgeleme problemleri NP-zor sınıfta yer almaktadır (Xing and Zhang, 2000). Bir makinede n tane işe ait alt işler $n!$ sayıda sıralanabilmektedir. Bu bir makinedeki sıralamanın herhangi bir makineye bağımlılık göstermediği, alt işleri oluşturan birim iş sayısının en fazla o işe ait toplam birim iş sayısı kadar değer alabildiği düşünülürse uygun olmayan ya da kısıtları sağlamayanlarla birlikte $(n!)^m(u+1)^{nm}$ sayıda olası çizelge oluşturulabilmektedir.

Problemimize ait matematiksel model incelendiğinde; $3n^2m+2nm+n$ sayıda kısıtımızın, n^2m tanesi 0-1 tamsayılı, n^2m tanesi sıfırdan büyük tamsayılı, n^2m+n tanesinin sıfırdan büyük ya da eşit reel sayılı değişkenlerimizin olduğu görülmektedir. Bu durumda her birinin 4 birim işe sahip olduğu 4 iş ve 2 makineyi içeren küçük boyutlu bir çizelgelemede bile toplam 116 adet kısıt; 32 adet 0-1 tamsayılı, 32 adet tamsayılı ve 36 adet reel sayılı değişkene ihtiyaç duyulacaktır. İş, makine ve birim iş sayılarındaki küçücük artışlarda bile modelin en iyi çözümü üretmesi daha da zor bir hâle gelecektir.

BÖLÜM 4

İŞLERİN BÖLÜNEBİLİR OLDUĞU, SIRADAN BAĞIMSIZ HAZIRLIK SÜRELERİNİ İÇEREN ÖZDEŞ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ İÇİN TABU ARAMA YÖNTEMİ

4.1 Tanım

Tabu, Tonga adasının yerli halkı tarafından kutsal sayılan ve dokunulması yasak olan nesnelere ifade etmek için kullanılan bir kelimedir. Webster Sözlüğü'ne göre ise, "töre tarafından konulan, koruyucu özellikteki yasak" anlamına gelmektedir (Glover and Laguna, 1997).

Tabu Arama (TA, Yasaklı Arama), kombinatoriyel eniyileme problemlerinin çözümünde yerel en iyi noktadan kaçınmak için kullanılan bir meta sezgisel metottur. İlk olarak 1986 yılında Fred Glover tarafından yeni bir çözüm yaklaşımı olarak ortaya atılmıştır. Tabu arama algoritması kullanılarak birçok kombinatoriyel eniyileme problemi için başarılı sonuçlar elde edilmiştir (Glover et al., 1993). Glover and Laguna'ya (1997) göre TA'nın uygulandığı başlıca kullanım alanları Çizelge 4.1'de gösterilmiştir.

4.2 Tabu Aramanın Esasları

Bir X kümesinde $f(x)$ fonksiyonu tanımlanmış olsun. Tabu Arama, belirlenmiş bir durdurma kriteri sağlanıncaya kadar ardıştırmalı olarak bir noktadan (çözümünden) diğerine geçen sıradan bir yerel arama algoritması gibi başlamaktadır. Her $x \in X$, $N(x) \subset X$ şeklinde bir komşuluğa sahiptir. x çözümünden bir $x' \in N(x)$ çözümüne gitmek için yapılan işlem, "hareket" olarak adlandırılmaktadır.

Çizelge 4.1. Tabu Aramanın uygulama alanları.

| | |
|--|--|
| <p>Çizelgeleme</p> <ul style="list-style-type: none"> • Hücreyel İmalat • Homojen olmayan İşlemci Çizelgeleme • İş gücü çizelgeleme • Derslik Çizelgelemesi • Makine Çizelgeleme • Akış Tipi Çizelgeleme • Atölye Tipi Çizelgeleme • Sıralama ve Partileme <p>Tasarım</p> <ul style="list-style-type: none"> • Bilgisayar Destekli Tasarım • Hataya Dayanıklı Ağlar • Ulaşım Ağı Tasarımı • Mimari Mekan Planlaması • Sabit Bedelli Ağ Tasarımı • Düzensiz Kesme Problemleri <p>Yerleşim ve Tahsis</p> <ul style="list-style-type: none"> • Çok ürünlü Yerleşim/Tahsis • Kareli Atama • Tesis Planlaması • Açık Deniz Petrolü Keşfi <p>Mantık ve Yapay Zeka</p> <ul style="list-style-type: none"> • Maksimum Sağlanabilirlik • Kümeleme • Örüntü Tanıma • Sınır Ağları/ Eğitim ve Tasarım <p>Teknoloji</p> <ul style="list-style-type: none"> • Elektrik Dağıtımı • Düşük Hacimli Elipsoit • Uzay İstasyonu İnşası • Devre Yerleşimi | <p>İletişim</p> <ul style="list-style-type: none"> • Çağrı Yönlendirme • Hizmet Ağı Tasarımı • Müşteri İndirimi Planlaması • Eş Zamanlı Optik Ağlar <p>Üretim, Stok ve Yatırım</p> <ul style="list-style-type: none"> • Esnek Üretim • Tam Zamanında Üretim • Kapasiteli MRP • Parça Seçimi • Çok Mamullü Stok Planlaması • Sabit Karma Yatırım <p>Rotalama</p> <ul style="list-style-type: none"> • Araç Rotalama • Kapasiteli Rotalama • Filo Rotalama • Gezgin Satıcı <p>Çizge Optimizasyonu</p> <ul style="list-style-type: none"> • Çizge Parçalama • Çizge Boyama • P-Medyan Problemleri <p>Genel Kombinatoriyel Optimizasyon</p> <ul style="list-style-type: none"> • Sıfır-Bir Programlama • Sabit Bedelli Optimizasyon • Doğrusal Olmayan Programlama • Karma Tamsayılı Programlama |
|--|--|

Arama ilerledikçe $N(x)$ üzerinde değişiklikler meydana getiren stratejilerin uygulanmasıyla daha etkili bir $N^*(x)$ komşuluğunun tanımlanması, TA'yı yerel aramanın ötesine taşımaktadır. TA'nın kilit noktası da $N^*(x)$ komşuluğunu oluşturmaya yardımcı olan bu özel hafıza yapılarının kullanımınıdır.

Bir TA, kısa ve uzun dönem hafızalı olmak üzere iki tipte ele alınmaktadır. TA'nın temelini kısa dönem hafızalı yapı oluşturmaktadır. Her ne kadar genel olarak uzun dönem hafızalı TA, problem çözümünde daha güçlü bir yapıya sahip olsa da kısa dönem hafızalı TA da yeterli düzeyde kaliteli sonuçlar elde etmektedir (Glover and Laguna, 1997).

4.3 Kısa Dönem Hafıza

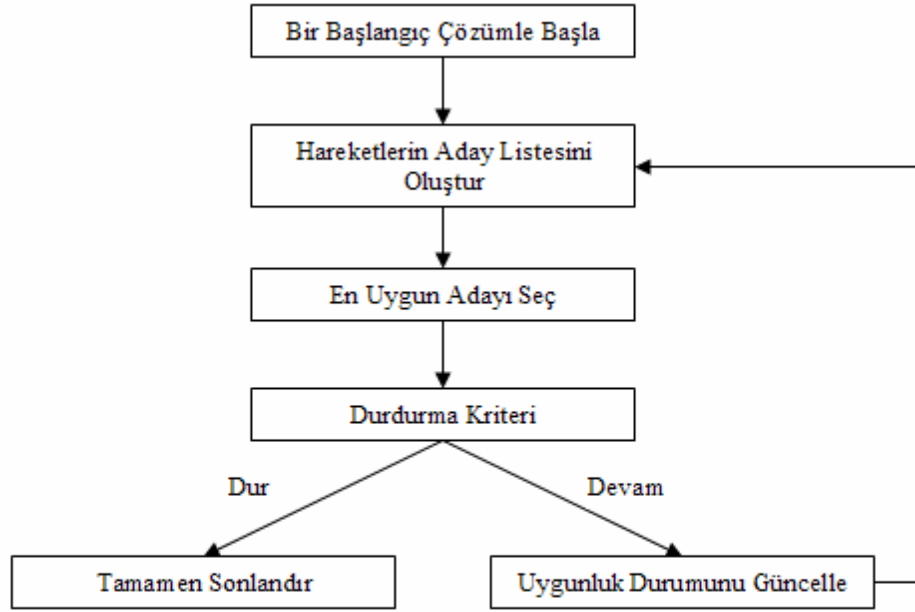
Önceden de belirtildiği gibi TA'nın temelini kısa dönem hafıza yapısı oluşturmakla beraber TA algoritmasını uygulamada bir takım stratejik değerlendirmeler yer almaktadır.

TA, bir başlangıç noktadan başlamakta ve komşu noktalar boyunca ilerlemektedir. Komşuluk yapısı, mevcut çözümlerden başka çözümler üretilmesine olanak sağlamaktadır. TA'da bütün komşuları dolaşmak büyük bir çaba gerektirdiğinden *aday listesi* kullanımına gidilmektedir. *Aday listesi*, belirli bir ardıştırmaya ait mevcut hareketlerin sınırlandırılarak oluşturulmuş bir alt kümesi olarak tanımlanabilir. Hareketlere ait aday listesi belirlendikten sonra aralarından en uygun aday seçilmektedir. Öncelikle bir hareketin tabu olup olmasına bakılmaksızın aday listedeki her bir hareket hesaplanmaktadır. Hesaplama mantığı olarak amaç fonksiyonundaki değişim (hareketi uygulamanın öncesindeki ve sonrasındaki amaç fonksiyonu değerleri farkı) göz önüne alınabilir. TA algoritmasında belirli hareketlere geri dönmeyi ya da belirli hareketlerin tekrar etmesini engelleyen bir takım kısıtlamalar mevcuttur. Tabu kısıtlamaları hareketlerin niteliklerine göre tanımlanmakta; tabu

hareketleri ise bu tabu kısıtlamalarına dayanmaktadır. Bu yöntem sayesinde çevrime neden olabilecek nitelikteki noktalar yasaklanarak (tabu hâline getirilerek) çözüm uzayında keşfedilmemiş noktalara doğru aramanın kaydırılması sağlanmaktadır. Bu durum TA'nın önemli bir özelliği olan *tabu listesi* sayesinde gerçekleştirilmektedir. Bir tabu listesi, yeni ziyaret edilmiş noktalara tekrardan gidilmemesi için yapılan en son hareketlerden oluşmaktadır. Tabu olarak etiketlenen bir hareket, bir sonraki T artırma boyunca tabu listesinde yer almaktadır. Buradaki T , tabu listesinin süresi olarak adlandırılmaktadır. Bir diğer önemli özellik ise *tabu yıkma kriteridir* (aspiration criterion). Esnekliğin sağlanması açısından tabu yıkma kriteri uygulanmaktadır. Uygunluk için bir hareketin tabu durumunun kontrol edilmesi ilk adımdır. Şayet bir hareket tabu değilse anında uygun hareket olarak kabul edilmekte; tabu ise, tabu durumunu yıkmasını sağlayacak olanağı *tabu yıkma kriteri* kendisine sağlamaktadır. Aday listesindeki diğer hareketlerden daha iyi sonuç veren harekete ait çözüm yeni çözüm olarak; şayet önceki en iyi çözümden de iyiyse en iyi çözüm olarak işaretlenmektedir. Bu durum son en iyi çözüm bulunana ya da belirli bir artırma sayısına ulaşılan kadar devam etmektedir (Glover and Laguna, 1997).

TA kısa dönem hafıza bileşenleri Şekil 4.1'de verilmiştir. Şekilde gösterildiği üzere en basit anlamda bir TA algoritması şu beş elemanı içermektedir:

- i. Bir sezgisel algoritmayla uygun bir başlangıç çözümün yaratılması.
- ii. Genel en iyiye ulaşmak için her bir artırmada çözümleri değiştirecek, dahası bir x çözümüne $N(x)$ komşuluğu oluşturacak hareket tipinin seçilmesi.
- iii. Tabu listesinin oluşturulması.
- iv. Tabu olduğu hâlde kayda değer bir iyileşme sağlayan bir hareketin durumunu değiştirecek olan tabu yıkma kriterinin belirlenmesi.
- v. Sonlandırma kriterinin belirlenmesi.

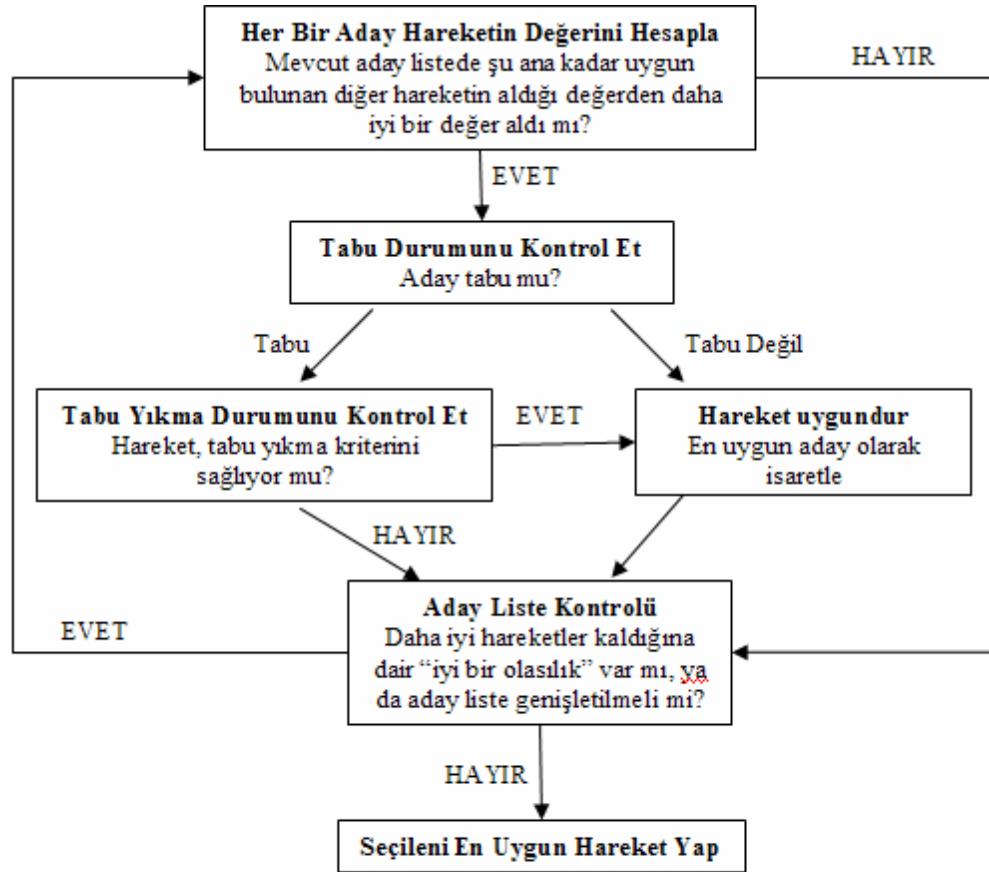


Şekil 4.1 Kısa dönem hafıza bileşenleri.

Bir başlangıç çözümüyle başlayan TA, sırasıyla hareketlerin aday listesinin oluşturulması ve aralarından en uygun adayın seçimiyle devam etmekte, bir durdurma kriterinin değerine göre de aramanın sonlandırılmasına ya da tekrarlanmasına karar vermektedir.

4.3.1 Komşuluk ve aday liste stratejileri

TA, bir başlangıç noktadan başlayıp, komşu noktalar arasında sırayla dolanmaktadır. Bu yönüyle TA, değişken bir komşuluk metodu olarak düşünülebilir. Bu, bir hareketin komşuluğunun statik bir kümeden ziyade aramanın geçmişine göre değişkenlik gösteren bir küme olduğu anlamına gelmektedir. Her bir adımda tabu durumları da kontrol edilerek bir sonraki çözümün seçileceği komşuluklar yeniden tanımlanmaktadır. En uygun adayın seçimi Şekil 4.2’de bir akış şemasıyla verilmiştir.



Şekil 4.2 En uygun adayın seçimi.

Kısa dönem hafızalı bir TA'da $N^*(x)$, tabu listesinin (T) dâhil edilmediği $N(x)$ 'in bir alt kümesidir; yani, $N^*(x) = N(x) - T$ dir. Diğer yandan uzun dönem hafızalı bir TA'da ise $N^*(x)$, genellikle $N(x)$ içerisinde yer almayan çözümleri de içerecek şekilde genişletilmektedir.

Tüm komşuluğun taranması genelde yüksek kalitedeki çözümlerin elde edilmesini sağlasa da hesaplamalar için harcanan süre açısından büyük bir yük getirmektedir. Mevcut çözüme ait olası tüm komşulukların hesaplanmaya

çalışılmasından ziyade daha akılcı prosedürler kullanarak bazı aday hareketlerinin izole edilmesi, verimlilik ve kaliteyi geniş ölçüde etkileyebilmektedir. Bu durum bilhassa, $N^*(x)$ 'ın büyük ve ona ait hareketlerin değerlerinin hesaplanmasının güç olduğu durumlar için geçerlidir. Bu yüzden, arama sürecinde arzu edilir özellikteki hareketleri içeren komşuluk bölgelerini izole edecek ve bunları o anki arama esnasında aday listesine koyacak etkili aday liste stratejilerinin belirlenmesi önem arz etmektedir.

Aday liste prosedürü, tüm komşuluğun bir alt kümesini oluşturmaya yarayan ve bu alt kümenin büyüklüğünü kontrol eden bir prosedür olarak tanımlanabilir. Büyüklüğün belirlenmesi önemlidir. Genelde komşuluk büyüdükçe, yerel en iyiye takılma olasılığı azalmaktadır. Ancak, büyük komşulukları taramanın maliyeti genellikle çok yüksektir. Bu yüzden aday liste stratejisi, yalnızca tek bir ardıştırmada hesaplama gücünü azaltıcı şekilde değil, aynı zamanda da istenen süre zarfında kaliteli çözümler bulabilecek şekilde belirlenmelidir (Glover and Laguna, 1997).

4.3.2 Tabu listesi

Geleneksel yerel arama metodlarının dezavantajı, bir yerel en iyiye takılmaları, böylece aramayı devam ettirememeleridir. Bu da kullanıcıyı, yerel en iyiden kaçınmayı sağlayan sezgisellere yönlendirmektedir. TA, iyileşme sağlamayan hareketlere izin vererek yerel en iyinin üstesinden gelmeye çalışmaktadır. Aramanın her defasında bazı çözümler arasında gidip gelmesini engellemek amacıyla önceden ziyaret edilmiş komşuluklara giden hareketlerin, yani tabu hareketlerin listesi çıkarılmaktadır. Buna *tabu listesi* adı verilmektedir. Tabu listesi, en son yapılan hareketlerden meydana gelmektedir; böylece, algoritmanın çevrimi önlemesi ve aramayı keşfedilmemiş bölgelere yönlendirmesi sağlanmaktadır (Glover, 1990).

Yeniliğe dayalı hafıza yapısına sahip olan tabu listesinde, belirlenmiş bir hareketin niteliğinin kaç ardıştırmaya boyunca tabu olarak kalacağını *tabu süresi* (tabu listesi uzunluğu - tabu tenure) belirlemektedir. Tabu süresinin, TA'nın performansını

etkilediği bilinmektedir. Tabu süresi kısa tutulduğunda, TA durmadan aynı yerel en iyiye dönebilir. Bu durum, aramayı daha geniş alanlara taşımayı engellemektedir. Uzun tutulduğunda ise, bir hareketin tabu olup olmadığının belirlenmesi için o hareketin tabu listesinde aranması daha da uzun sürecektir. Böylece algoritmanın çalışma süresi boyunca çözüm uzayını taraması için harcadığı süre azalacaktır (Glover et al., 1993).

Önceki birçok TA uygulamaları dikkate alındığında, en iyi tabu süresinin 5 ila 12 arasında değerler aldığı söylenebilir. Literatürde en çok tabu süresinin 7 alındığı durumla karşılaşılmaktadır. Bir tabu süresi, sabit ya da değişken nitelikte olabilir. Bir çok uygulamada hem başarılı bir sonuç vermesi hem de kullanılışının basitliği açısından statik tabu süresi ele alınmaktadır. Böyle kullanıldığı durumda hareketler, belirlenmiş bir artırma sayısı kadar tabu olarak kalmaktadır. Diğer yandan, bir hareketin niteliğine göre tabu durumunun değişkenlik gösterdiği dinamik tabu süresine de çalışmalarda yer verilmektedir (Dammeyer and Voss, 1993).

4.3.3 Tabu yıkma kriteri

Bir tabu durumunun tanımlanması, bir yandan çevrimi önlerken bir yandan da sanki ziyaret edilmemiş çözümlere yönelmeyi sağlayan hareketleri yasaklayan bir tutum içinde olduğu hissini vermektedir. Bunun için bazı durumlarda hareketlerin tabu durumlarında değişikliğe gidilmesi kaçınılmaz olmaktadır. Tam bu noktada *tabu yıkma kriteri* devreye girmektedir. Tabu olduğu hâlde tabu yıkma kriterini sağlayan bir hareket, uygun bir hareket olarak aday listesindeki yerini almaktadır.

Uygunluk için ilk gerekli adım, bir hareketin tabu durumunun kontrol edilmesidir (Şekil 4.2). Şayet bir hareket tabu değilse anında uygun olarak kabul edilmekte; aksi hâlde harekete uygunluk için tabu yıkma kriteriyle ikinci bir şans sunulmaktadır. Böylece, yukarıda değinilmeye çalışılan daha iyi çözümlere gitme yönünde bir esneklik sağlanmış olmaktadır (Glover, 1989).

Literatürdeki birçok uygulamada tabu yıkma kriteri iki şekilde ele alınmaktadır. Bunlardan ilki, amaca dayalı bir kriter olup, o ana kadar bulunan en iyi sonuçtan daha iyi bir sonuç vermesi durumunda ilgili tabu hareketin kabul edilmesi yönünde; diğeri ise yokluğa dayalı olup, tüm olası hareketlerin tabu olduğu durumda tabu süresinin bitmesi en yakın olan bir hareketin seçilmesi yönündedir.

4.3.4 Tabu Arama algoritması

Enküçükleme problemleri için temel bir TA algoritması şu adımlardan oluşmaktadır:

Adım 1.

$$k = 1$$

Bir takım sezgisel yöntemlerle başlangıç bir çözüm, S_1 , elde et; en iyi çözümü bu çözüme eşitle, $S_{eniyi} = S_1$.

Adım 2.

İlgili çözüme ait bir komşu seç, $S_c \in N(S_k)$

Şayet $S_k \rightarrow S_c$ tabu listesinde yer alan bir hareket tarafından yasaklanmışsa,

Adım 2'ye geri dön

Şayet $S_k \rightarrow S_c$ tabu listesinde yer alan bir hareket tarafından yasaklanmamışsa,

$$S_{k+1} = S_c \text{ yap}$$

Hareketin tersini tabu listesinin başına yaz

Tabu listesindeki diğer tüm girdileri bir sıra aşağıya çek

Tabu listesinin en altında yer alan girdiyi sil

Şayet $F(S_c) < F(S_{eniyi})$ ise $S_{eniyi} = S_c$ yap

Adım 3'e git.

Adım 3.

$$k = k+1$$

Şayet durdurma kriteri = doğru ise DUR; değilse, *Adım 2*'ye git.

Genel bir TA'nın uygulanışını örnek bir problem üzerinde görmek yarar sağlayacaktır. Örnek problemimizde $1|d_j|\sum w_j T_j$ çizelgeleme problemi ele alınsın. Probleme dair veriler, Çizelge 4.2'de verildiği gibi olsun.

Çizelge 4.2 Örnek probleme dair veriler.

| İşler | 1 | 2 | 3 | 4 |
|-------|----|----|----|----|
| p_j | 10 | 10 | 13 | 4 |
| d_j | 4 | 2 | 1 | 12 |
| w_j | 14 | 12 | 1 | 12 |

Komşuluk yapısının, ardışık iki işin yer değiştirdiği; tabu süresinin 2 olup, tabu listesinde yer değiştiren (j,k) iş çiftlerine yer verildiği; durdurma kriterinin ise 4 ardıştırma ile sınırlı olduğu bir TA uygulaması ele alınsın. Bu bilgiler dâhilinde TA şu şekilde gerçekleşecektir:

$$S_1 = 2, 1, 4, 3$$

$$F(S_1) = \sum w_j T_j = 12*8 + 14*16 + 12*12 + 1*36 = 500 = F(S_{eniye})$$

$$F(1, 2, 4, 3) = 480$$

$$F(2, \underline{4}, \underline{1}, 3) = 436 = F(S_{eniye})$$

$$F(2, 1, 3, 4) = 652$$

Tabu listesi: $\{(1,4)\}$

$$S_2 = 2, 4, 1, 3 \quad F(S_2) = 436$$

$$F(\underline{4}, \underline{2}, 1, 3) = 460$$

$$F(2, 1, 4, 3) (= 500) \quad \text{tabu!}$$

$$F(2, 4, 3, 1) = 608$$

Tabu listesi: $\{(2,4),(1,4)\}$

$$S_3 = 4, 2, 1, 3 \quad F(S_3) = 460$$

$$F(2, 4, 1, 3) (= 436) \quad \text{tabu!}$$

$$F(4, \underline{1}, \underline{2}, 3) = 440$$

$$F(4, 2, 3, 1) = 632$$

$$\text{Tabu listesi: } \{(2,1),(2,4)\}$$

$$S_4 = 4, 1, 2, 3 \quad F(S_4) = 440$$

$$F(\underline{1}, \underline{4}, 2, 3) = 408 \quad = F(S_{eni\ddot{y}i})$$

$$F(4, 2, 1, 3) (= 460) \quad \text{tabu!}$$

$$F(4, 1, 3, 2) = 586$$

$$\text{Tabu listesi: } \{(4,1),(2,1)\}$$

$$F(S_{eni\ddot{y}i}) = 408$$

4.4 Uzun Dönem Hafıza

Kısa dönem hafıza, yakın geçmiştekilerin etkisini yok edecek hareketlerden sakınmak için kullanılmaktadır. Yani genel olarak, bir kısa dönem hafıza fonksiyonu, yakın geçmişte karşılaşılan durumların tekrarını önlemeye hizmet etmektedir. Ancak bazı durumlarda tek başına, aramayı daha iyi çözümlerin yer aldığı bölgelere taşımakta yetersiz gelmekte, en iyi sonuca giden doğru hareketi yakalamada başarısız olmaktadır. Bu da yeni bölgelere doğru aramanın çeşitlendirilmesini (diversification) yetersiz kılmaktadır. Uzun dönem hafızanın tamamlayıcı yönü burada kendini göstermektedir. Bu sayede uzun dönem hafızayı ve onun bileşenlerini içeren bir TA, aramayı daha da güçlendirmektedir.

Orta ve uzun dönem hafıza başlıca, aramayı şiddetlendirmede (intensification) ve çeşitlendirmede (diversification) kullanılan stratejilere temel teşkil etmektedir. TA içinde uygulanan orta ve uzun dönem hafıza fonksiyonları, bölgesel şiddetlendirmeyi (regional intensification) ve genel çeşitlendirmeyi (global diversification) gerçekleştirmek amaçlıdır. Orta dönem hafıza, daha ziyade daha önce bulunan iyi

çözümleri barındıran bölgelere yoğunlaşırken; uzun dönem hafıza ise aramayı, daha önce keşfedilmemiş bölgelere doğru çeşitlendirmektedir.

Uzun dönem hafızada sıklık (frequency) bilgileri önemli bir rol oynamaktadır. Sıklık bilgisinin ardında yatan düşünce, geçmişte aynı tercihlerin ne kadar sıklıkta yapıldığından faydalanarak yerel tercihleri yönlendirmektir. İki tip sıklık mevcuttur. İlki, elemanların ne kadar sıklıkta çözümün bir parçası olduğunu izleyen sıklık (residence measure); diğeri ise, elemanların ne kadar sıklıkta değişikliğe uğradığını izleyen sıklıktır (transition measure).

4.5 İşlerin Bölünebilir Olduğu, Sıradan Bağımsız Hazırlık Sürelerini İçeren Özdeş Paralel Makine Çizelgeleme Problemi İçin Tabu Arama Yöntemi

Çalışmada, ele alınan özdeş paralel makine problemi için kısa dönem hafızalı Tabu Araması yaklaşımı önerilmektedir. İlgili probleme dair bilinen tek TA uygulaması olması da dikkate alınarak, izleyen alt başlıklarda öncelikle önerilen TA'na dair tanımlamalarda bulunulmuş, parametre değerlerinin uygun seçimleri deney tasarımıyla aranmaya çalışılmış, daha sonra önerilen TA, çeşitli test problemleri üzerinde ilgili problem için kurulan matematiksel modelin Lingo yazılımında çözdürülmesiyle elde edilen sonuçlarla karşılaştırılmıştır.

Önerilen TA'na dair çözümün kodlanması, başlangıç çözümün elde edilmesi, komşuluk ve aday liste stratejileri, tabu listesinin oluşturulması, tabu süresi, tabu yıkma kriterini içeren tanımlamalara izleyen kısımda değinilmektedir. Sırasıyla tanımlamalar şu şekilde yapılmaktadır:

Çözümün kodlanması; makine sayısı, m , kadar kısmi çizelgeyi içerecek şekilde

$$\begin{aligned}
\text{Makine 1:} & \left[J_1, U_{J_1} \right] \left[J_2, U_{J_2} \right] \dots \dots \left[J_{n-1}, U_{J_{n-1}} \right] \left[J_n, U_{J_n} \right] \\
\text{Makine 2:} & \left[J_1, U_{J_1} \right] \left[J_2, U_{J_2} \right] \dots \dots \left[J_{n-1}, U_{J_{n-1}} \right] \left[J_n, U_{J_n} \right] \\
& \cdot \\
& \cdot \\
\text{Makine m:} & \left[J_1, U_{J_1} \right] \left[J_2, U_{J_2} \right] \dots \dots \left[J_{n-1}, U_{J_{n-1}} \right] \left[J_n, U_{J_n} \right]
\end{aligned}$$

olarak ele alınmıştır.

$\left[J_1, U_{J_1} \right]$ sırasıyla, ilgili makinenin 1. pozisyonunda yer alan alt işi ve o alt işin birim iş sayısını; $\left[J_2, U_{J_2} \right]$ ilgili makinenin 2. pozisyonunda yer alan alt işi ve o alt işin birim sayısını gösterecek şekilde devam ederek; nihayetinde ilgili makinenin n. pozisyonunda yer alan alt işi ve o alt işin birim sayısını gösteren $\left[J_n, U_{J_n} \right]$ ile son bulmaktadır.

Başlangıç çözüm olarak; amaç fonksiyonu dikkate alındığında, TA için başlangıç çözüm üretmede çokça yer verilen En Erken Teslim Zamanı (Earliest Due Date-EDD) sevk etme kuralı kullanılmıştır. Önce, EDD kuralıyla işler sıralanmış; sonra da özdeş paralel makinelere atamaları sırasıyla yapılmıştır.

Komşuluk yapısında genel ikili yer değiştirme (general pairwise interchange) ve ekle çıkar (insert) hareketlerinden oluşan melez bir hareket tipi ele alınmıştır. Genel anlamda bir ekle çıkar hareket için iki iş tanımlanmakta ve ilk iş, ikinci işin hemen önüne gelecek şekilde yerleştirilmektedir. Yer değiştirme hareketinde ise işler, birbirlerinin önceki yerlerine yerleştirilmektedir. Paralel makine çizelgeleme problemlerinde ilgili yerler, aynı makinede olabileceği gibi farklı makinelerde de olabilmektedir.

Önerilen TA’da ekle çıkar hareketi, farklı iki makinede yer alan aynı işe ait alt işlerin birim iş sayıları üzerinde gerçekleştirilmektedir. Uygun birinde bir birim iş sayısı kadar çıkarma, diğerinde ise bir birim iş sayısı kadar artış yapılmaktadır. Böylelikle ilgili işe ait toplam birim iş sayısının korunması da sağlanmış olmaktadır. Bu, önerilen TA için ekle çıkar hareketini oluşturmaktadır. Yer değiştirme hareketi ise yalnızca aynı makinede yer alan işler arasında gerçekleştirilmektedir. Yer değiştirme hareketinin farklı makinelerde gerçekleştirilmesine izin verilmemesinin nedeni, çözüm kodlaması gereğince aynı makinede aynı işe ait birden fazla alt işin çizelgelenmesine; başka bir deyişle, kurulan matematiksel modelin (2.2) kısıtını sağlamayıp uygun çözüm alanından çıkmasına olanak sağlamasıdır.

Bir çözüme ait, tanımlanan ikili yer değiştirme hareketlerine nazaran ekle çıkar hareketlerinin oluşturduğu komşuluk büyük olduğundan, *aday liste stratejisi* yalnızca ekle çıkar hareketleri üzerinde uygulanmaktadır. Bunun için, bir taraftan tüm komşulukları hesaplama gücünü azaltabilmek, bir taraftan da istenen süre zarfında kaliteli çözümler bulabilmek adına iki tane aday liste stratejisi sınanmaktadır. Biri, Bilge et. al (2004) çalışmasında yer alan, her bir makineye ait gecikme hesaplanarak $(\sum_{i \in I_j} T_i, I_j: \text{makine } j\text{'de çizelgelenen işlerin kümesi})$, “toplam gecikmeye en fazla katkıda bulunan makine”nin seçimine dayanıp, bu makineden diğer tüm makinelere ekle çıkar hareketinin yapıldığı aday liste stratejidir. Diğer ise, Kim et al. (2004) çalışmasında makine seçimi kuralları içerisinde yer alan, minimum iş yüküne (ilgili makineye atanan işlerin hazırlık sürelerinin ve işlem sürelerinin toplamı) sahip makinenin seçimine dayanıp diğer makinelerden bu makineye ekle çıkar hareketinin tanımlandığı aday liste stratejisidir.

Tabu kısıtlamalarını oluşturmada Çizelge 4.3’te gösterilen kısıtlamalar arasından hem ikili yer değiştirme hem de ekle çıkar hareket yapısı için iki ayrı nitelik belirlenmiş, bunlar önerilen TA yapısına uyarlanmıştır.

Çizelge 4.3. İkili yer deęiřtirme için nitelikler ve tabu kısıtlamaları (Laguna et al., 1991; Bilgen'den, 1999).

| Nitelik | Tabu Kısıtlaması |
|------------------------|--|
| $(İş\ i, İş\ j, i, j)$ | 1. İş i 'nin i . sırada ve iş j 'nin j . sırada yer aldığı bir hareketi engelle. |
| | 2. İş i 'nin i . sırada ya da iş j 'nin j . sırada yer aldığı bir hareketi engelle. |
| $(İş\ i, i)$ | 3. İş i 'nin i 'ye dönmesini engelle. |
| | 4. İş i 'nin k 'ya, $k \leq i$ hareketini engelle. |
| $(İş\ i, j)$ | 5. İş i 'nin k 'ya, $k \leq j$ hareketini engelle; yani řu an bulunduğu yerden önce gelen yerlere olan hareketi engelle. |
| $(İş\ i)$ | 6. İş i 'nin hareketini engelle. |
| $(İş\ i, İş\ j)$ | 7. İş i 'nin ve iş j 'nin hareketini engelle. |

Ekle çıkar hareketi için tabloda yer alan 6 no'lu tabu kısıtlaması, makine k 'daki alt iş i 'nin çıkar hareketini engelle, (makine k , alt iş i) řeklinde; ikili yer deęiřtirme hareketi için ise 5 no'lu tabu kısıtlaması, makine k 'da alt iş i 'yi, o alt işe ait birim iş sayısı u_i 'den büyük veya eřit olduęu müddetçe, bulunduğu yerden önce gelen yerlere yapılan hareketleri engelle, (makine k , alt iş i , u_i, j) řeklinde uyarlanmıřtır.

İlgili farklı iki tabu özellięini tutan iki tabu listesi için *tabu süreleri* arama boyunca tek bir deęerde sabitlenmiřtir. *Tabu yıkma kriteri*'nde amaç fonksiyonu düşünölmüş olup ilgili kritere göre, tabu olan bir hareket, mevcut ardıřtırmaya kadar bulunmuş en iyi çözümden daha iyi bir çözümün elde edilmesini saęlıyorsa, tabu olmasına raęmen gerçekleştirilmiřtir.

Durdurma kořulu, algoritma mevcut en iyi çözümden önceden belirlenmiş bir ardıřtırma sayısı süresince bir iyileřme saęlanamadığı takdirde uygulanmıřtır.

4.6 Deneysel Çalışmalar

Önerilen Tabu Aramanın çözüm kalitesi üzerinde tabu aramasına ait yukarıda bahsedilen parametrelerin etkinliği araştırılıp, en iyi sonucun alındığı parametre seti belirlenmek istenmiştir. Tüm deneysel çalışmalarda Shim and Kim'in (2008) çalışmasında yer alan veri setinden yararlanılmıştır. Problem büyüklükleri, makine, iş ve birim iş sayılarıyla tanımlanmış olup, çalışma için sırasıyla makine sayıları 4, 8, 12; iş sayıları 10, 15, 20, 25 ve birim iş sayıları 4, 6, 8, 10 olarak alınmıştır. Shim and Kim'in (2008) çalışmalarında yer verdiği problem setleri dikkate alınarak; bunlardan makine sayıları, iş sayıları ve birim iş sayıları sırasıyla (4, 10, 4), (4, 15, 6) ve (8, 15, 6) olanlar Test Problemleri-I; (8, 20, 8), (12, 20, 8) ve (12, 25, 10) olanlar ise Test Problemleri-II olarak gruplandırılmıştır. Hazırlık süreleri için kısa, orta ve uzun olmak üzere üç düzey; teslim zamanlarını oluşturmak için ise dört farklı düzey veri seti kullanılmıştır. Diğer veriler, gerçek problemleri nispeten iyi yansıtacak şekilde şöyle oluşturulmuştur (Shim and Kim, 2008):

- i. Bir birim işin işlem süresi, [5,60] arasında değişen kesikli düzgün dağılımdan oluşturulmuştur.
- ii. Bir iş için hazırlık süresi, sırasıyla kısa, orta ve uzun hazırlık süresi olacak şekilde [5,60], [60,120] ve [120,180] arasında değişen kesikli düzgün dağılımdan elde edilmiştir.
- iii. Teslim zamanı ise Kim et al. (2004) çalışmasında yer verdiği şekilde oluşturulmuştur. Teslim zamanı, $\left[\alpha \times \sum_i (s_i + u_i p_i) / m, \beta \times \sum_i (s_i + u_i p_i) / m, \right]$ arasında değişen kesikli düzgün dağılımdan elde edilmiştir. s_i , p_i , u_i ve m sırasıyla i işine ait hazırlık süresini, bir birim işin işlem süresini, birim iş sayısını ve makine sayısını göstermektedir.

Çalışmada; $(\alpha, \beta) = (0, 1.2)$ alınmış, hazırlık süresi kısa tutulmuştur. Test problemleri, kurulan matematiksel model dâhilinde Lingo programı yardımıyla

çözdürülmüştür. Kurulan matematiksel modelin Lingo programında yazımı EK-1’de verilmiştir. Tabu Arama ve test problemlerinin elde edilmesi ise TurboPascal programı ile kodlanmıştır. Test problemlerinin elde edilmesini içeren TurboPascal kodları EK-2’de; önerilen TA’ya ait başlangıç çözümün elde edilmesini, tabu listelerinin oluşturulmasını ve bir hareketin tabu olup olmadığının kontrol edilmesini içeren TurboPascal kodları ise sırasıyla EK-3, EK-4 ve EK-5’te verilmiştir. Tüm bilgisayar koşulları Intel Duo T2300 GHz işlemcili bir bilgisayarda gerçekleştirilmiştir.

4.6.1 Tabu Arama parametrelerinin belirlenmesi

Önerilen Tabu Arama için uygun tabu arama parametrelerinin belirlenmesi amacıyla deney tasarımı Test Problemleri-I ve Test Problemleri-II için ayrı ayrı yapılmıştır. Test Problemleri-I ve Test Problemleri-II için deney tasarımında göz önüne alınan faktör ve düzeyleri Çizelge 4.4’te gösterilmiştir.

Çizelge 4.4 Test Problemleri-I ve II için deney tasarımda göz önüne alınan faktör ve düzeyleri.

| | Faktör | Düşük Düzey (-1) | Orta Düzey (0) | Yüksek Düzey (+1) |
|----------------------------|------------------------------|-------------------------|-----------------------|--------------------------|
| Test Problemleri-I | Aday Liste Stratejisi | Makine İş Yüğü | - | Makine Gecikme |
| | Tabu Süresi (Yer Değiştirme) | 12 | 20 | 25 |
| | Tabu Süresi (Ekle Çıkar) | 7 | 12 | 15 |
| | Ardıştırma Sayısı | 1500 | - | 5000 |
| Test Problemleri-II | Aday Liste Stratejisi | Makine İş Yüğü | - | Makine Gecikme |
| | Tabu Süresi (Yer Değiştirme) | 12 | 20 | 25 |
| | Tabu Süresi (Ekle Çıkar) | 12 | 15 | 20 |
| | Ardıştırma Sayısı | 5000 | - | 8000 |

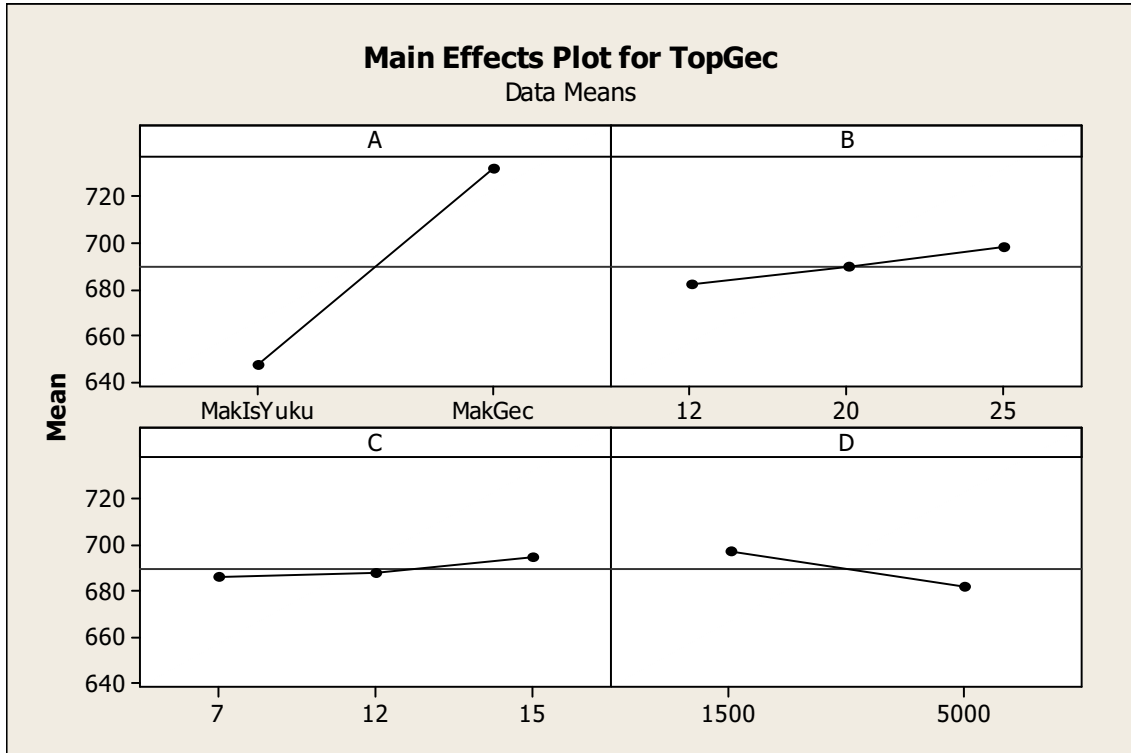
Önerilen Tabu Aramanın performansını önemli ölçüde etkileyebilecek dört faktör seçilmiştir: (A) aday liste stratejisi, (B) ikili yer değiştirme tabu listesi için tabu süresi, (C) ekle çıkar tabu listesi için tabu süresi ve (D) artırma sayısı (durdurma kriteri). Her iki test problemlerinde aday liste stratejisine ait düzeyler, bir önceki bölümde tanımlanan iki strateji (En Az Makine İş Yüğü, En Çok Geciken Makine); ikili yer değiştirme tabu listesi için tabu süresine ait düzeyler ise (12, 20, 25) olarak belirlenmiştir. Diğer iki faktöre (C ve D) ait düzeyler her iki test problemleri için ayrıdır. Test Problemleri-I'e ait ekle çıkar tabu listesi için tabu süresi düzeyleri (7, 12, 15) ve artırma sayısı düzeyleri (1500, 5000) alınırken; Test Problemleri-II'ye ait ekle çıkar tabu listesi için tabu süresi düzeyleri (12, 15, 20) ve artırma sayısı düzeyleri (5000, 8000) alınmıştır. Artırma sayısına ait düzeylerin belirlenmesinde Bilge et al. (2004) çalışmasından faydalanılmıştır. Tabu süreleri düzeyleri için ise benzer problemi ele alan başka bir meta sezgisel yaklaşım bulunmadığından önerilen Tabu Aramanın yapısına uygun olarak düzeyler belirlenmiştir.

Aday liste stratejisi ve artırma sayısı 2 düzey, diğer faktörler 3 düzeye sahiptir. Test Problemleri-I ve II'de tam faktöriyel deney için $2^2 \times 3^2 = 36$ farklı deney kombinasyonu mevcuttur. Her kombinasyon Test Problemleri-I ve II'deki 3'er probleme 3'er kez yinelenerek uygulanmıştır. Deneyler esnasında problem etkisini dikkate alarak açıklanamayan hatayı azaltmak amacıyla problem bazında bloklamaya gidilmiştir. Böylece Test Problemleri-I ve II için $36 \times 3 \times 3 = 324$ olmak üzere toplamda 648 deney yapılmıştır. Test Problemleri-I için Varyans Analizi sonuçları Çizelge 4.5'te verilmiştir.

Çizelge 4.5 Test Problemleri-I için Varyans Analizi sonuçları.

| Source | DF | Seq SS | Adj SS | Adj MS | F | P |
|--------|-----|----------|----------|---------|---------|-------|
| Blocks | 8 | 42014978 | 42014978 | 5251872 | 2100,34 | 0,000 |
| A | 1 | 593071 | 593071 | 593071 | 237,18 | 0,000 |
| B | 2 | 12937 | 12937 | 6468 | 2,59 | 0,077 |
| C | 2 | 4600 | 4600 | 2300 | 0,92 | 0,400 |
| D | 1 | 19818 | 19818 | 19818 | 7,93 | 0,005 |
| A*B | 2 | 9428 | 9428 | 4714 | 1,89 | 0,154 |
| A*C | 2 | 3974 | 3974 | 1987 | 0,79 | 0,453 |
| A*D | 1 | 499 | 499 | 499 | 0,20 | 0,655 |
| B*C | 4 | 3826 | 3826 | 957 | 0,38 | 0,821 |
| B*D | 2 | 1706 | 1706 | 853 | 0,34 | 0,711 |
| C*D | 2 | 820 | 820 | 410 | 0,16 | 0,849 |
| Error | 296 | 740144 | 740144 | 2500 | | |
| Total | 323 | 43405801 | | | | |

%5 anlam düzeyinde Test Problemleri-I için A ve D faktörleri kritiktir. Faktör düzeylerini belirlemek amacıyla ana etkiler grafiği çizilmiştir (Şekil 4.3).



Şekil 4.3 Test Problemleri-I için ana etkiler grafiği.

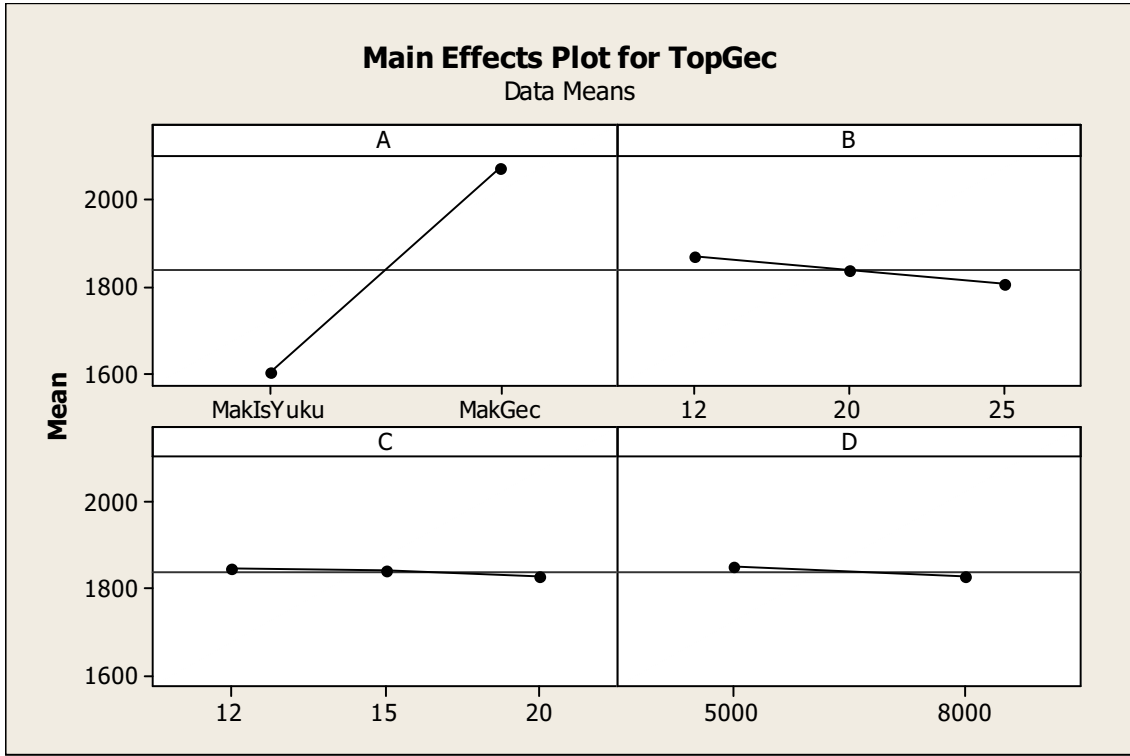
Amaç, en küçükleme olduğu için Test Problemleri-I aday liste stratejilerinden makine iş yükü en az olan makinenin seçimini içeren aday liste stratejisi ve durdurma kriteri olarak da 5000 değeri seçilmiştir.

Test Problemleri-II için ise Varyans Analizi sonuçları, Çizelge 4.6'de verilmiştir.

Çizelge 4.6 Test Problemleri-II için Varyans Analizi sonuçları.

| Source | DF | Seq SS | Adj SS | Adj MS | F | P |
|--------|-----|-----------|-----------|----------|---------|-------|
| Blocks | 8 | 222270682 | 222270682 | 27783835 | 1285,22 | 0,000 |
| A | 1 | 17992679 | 17992679 | 17992679 | 832,30 | 0,000 |
| B | 2 | 215506 | 215506 | 107753 | 4,98 | 0,007 |
| C | 2 | 19522 | 19522 | 9761 | 0,45 | 0,637 |
| D | 1 | 38025 | 38025 | 38025 | 1,76 | 0,186 |
| A*B | 2 | 36008 | 36008 | 18004 | 0,83 | 0,436 |
| A*C | 2 | 39378 | 39378 | 19689 | 0,91 | 0,403 |
| A*D | 1 | 232 | 232 | 232 | 0,01 | 0,918 |
| B*C | 4 | 81349 | 81349 | 20337 | 0,94 | 0,441 |
| B*D | 2 | 3243 | 3243 | 1621 | 0,07 | 0,928 |
| C*D | 2 | 668 | 668 | 334 | 0,02 | 0,985 |
| Error | 296 | 6398921 | 6398921 | 21618 | | |
| Total | 323 | 247096211 | | | | |

%5 anlam düzeyinde Test Problemleri-II için A ve B faktörleri kritiktir. Faktör düzeylerini belirlemek amacıyla ana etkiler grafiği çizilmiştir (Şekil 4.4).



Şekil 4.4 Test Problemleri-II için ana etkiler grafiği.

Amaç, en küçükleme olduğu için Test Problemleri-II aday liste stratejilerinden makine iş yükü en az olan makinenin seçimini içeren aday liste stratejisi ve yer değiştirme tabu listesi için tabu süresi 25 olarak seçilmiştir.

4.6.2 Önerilen Tabu Aramanın test problemleri üzerindeki performansı

Önerilen Tabu Aramanın test problemleri üzerindeki performansını görmek amacıyla en iyi çözümlerle karşılaştırma yoluna gidilmiştir. Ele alınan probleme dair en iyi sonucu veren bir algoritma var olmadığından, kurulan matematiksel modelin Lingo programında koşturulmasıyla en iyi sonuçlar elde edilmeye çalışılmıştır. Normal boyuttaki bir probleme en iyi çözümü bulmak uzun zaman gerektirdiğinden, testler küçük problemler üzerinde 10'ar yinelenmeyle denenmiştir. Deneyler yapılmadan önce Test Problemleri-I için elde edilen ana etkiler grafiklerinden yararlanılarak parametre değerleri girilmiştir. Kritik olmayan parametrelerde amaca katkısı olan değerler, yani

en küçük değerleri veren düzeyler seçilmiştir. Bu yüzden kritik olmayan (B) ve (C) parametrelerine sırasıyla 12 ve 7 değerleri girilmiştir. Kritik olan (A) parametresi için aynı düzey seçilirken, deney yapılan problemin küçüklüğü dikkate alınarak, durdurma kriteri değeri 1500 artırma seçilmiştir. Her bir probleme ait Tabu Aramanın ve Lingo programının verdiği sonuçlar EK-6'da verilmiştir. Alınan özet sonuçlar, Çizelge 4.7'de gösterilmiştir.

Çizelge 4.7. Küçük test problemleri üzerinde, önerilen Tabu Aramanın en iyi çözümlerden sapma yüzdeleri.

| Mak. Say. | İş Say. | Br. İş Say. | Problem Sayısı | En iyi çözümden sapma yüzdeleri | | TA'nın bulduğu en iyi çözüm sayısı | CPU Süre (sn) | | | |
|-----------|---------|-------------|----------------|---------------------------------|-----|------------------------------------|---------------|-----------|------|-----------|
| | | | | | | | Lingo | | TA | |
| | | | | | | | Ort. | Std. Sap. | Ort | Std. Sap. |
| 2 | 2 | 5 | 10 | 0.0 | 0.0 | 10 | 0.0 | 0.00 | 1.25 | 0.48 |
| 2 | 3 | 5 | 10 | 0.0 | 0.0 | 10 | 0.5 | 0.53 | 1.24 | 0.46 |
| 2 | 4 | 5 | 10 | 0.0 | 0.0 | 10 | 70.9 | 48.1 | 1.04 | 0.07 |
| 3 | 3 | 5 | 10 | 0.3 | 0.9 | 9 | 84.1 | 62.7 | 1.21 | 0.28 |
| 3 | 4 | 5 | 10 | 0.0 | 0.0 | 10 | 4871.2 | 3131.4 | 1.20 | 0.30 |

Çizelge 4.7 incelendiğinde ele alınan küçük boyutlu problemler üzerinde önerilen Tabu Aramanın, Lingo programından elde edilen sonuçlar ile hemen hemen aynı sonuçları daha kısa bir sürede elde ettiği görülmektedir. 50 problemin 49'unda eniyi sonucu yakalamıştır. Yakalanamayan problemde ise eniyi sonuçtan %3'lük bir sapma gerçekleşmiştir. Problem boyutu arttıkça Lingo programında en iyi çözüm elde etmek için gerekli zamanda ani bir artış meydana geldiği görülmektedir. Böyle olmasına rağmen, Tabu Aramanın yukarıdaki problem büyüklüklerinde her zaman Lingo ile aynı sonuçları vereceği garanti edilemez.

Önerilen Tabu Aramanın, Test Problemleri-I ve Test Problemleri-II üzerinde de performansı gözlenmiştir. TA parametre değerleri, yukarıda yapmış olduğumuz varyans analizlerinin sonuçları dikkate alınarak, Test Problemleri-I için kritik olan aday

liste stratejisi ve artırma sayısı parametreleri sırasıyla, “makine iş yükü” ve 5000 düzeylerine eşitlenmiş; kritik olmayan parametreler için de, amacımız en küçükleme olduğundan ana etkiler grafiğinde en küçük değerdeki düzeyler seçilmiştir. Neticede, Test Problemleri-I için aday liste stratejisi, “makine iş yükü”; yer değiştirme tabu süresi, 12; ekle çıkar tabu süresi, 7 ve artırma sayısı 5000 alınmıştır. Test Problemleri-II için ise aday liste stratejisi, “makine iş yükü”; yer değiştirme tabu süresi, 25; ekle çıkar tabu süresi, 20 ve artırma sayısı da 8000 alınmıştır.

Lingo koşulları için problem setlerindeki her bir problemin makul bir zaman diliminde en iyi çözüme ulaşamama durumuna karşı belli bir süre verilmiştir. Test Problemleri-I için bu süre, Shim and Kim’in (2008) çalışmasındaki gibi 3600 saniye olarak alınmıştır. Test Problemleri-II içinde yer alan problemlere Lingo yazılımı ilgili süre zarfında bir çözüm üretemediğinden bu süre 10800 saniyeye çıkartılmıştır. Elde edilen Lingo sonuçlarının Tabu Arama sonuçlarıyla karşılaştırılması Test Problemleri-I için Çizelge 4.8’de ve Test Problemleri-II için Çizelge 4.9’da verilmiştir.

Çizelge 4.8. Önerilen Tabu Aramanın Test Problemleri-I üzerindeki performansı.

| MS | IS | BrISS | Örn. No | Amaç Fonksiyonunun Değeri (Z) | | | | TA koşum zamanı (CPU sn.) | Ort. TA koşum Zamanı (CPU sn.) |
|--------------------------------------|----|-------|---------|-------------------------------|------|--------|---------------|---------------------------|--------------------------------|
| | | | | Lingo | TA | % Fark | % Fark (Ort.) | | |
| 4 | 10 | 4 | 1 | 850 | 579 | 31.88 | 32.03 | 5.98 | 5.13 |
| | | | 2 | 358 | 172 | 51.96 | | 5.39 | |
| | | | 3 | 1111 | 975 | 12.24 | | 4.01 | |
| 4 | 15 | 6 | 1 | 1815 | 199 | 89.04 | 76.87 | 5.33 | 5.62 |
| | | | 2 | 3155 | 862 | 72.68 | | 4.28 | |
| | | | 3 | 2141 | 666 | 68.89 | | 7.25 | |
| 8 | 15 | 6 | 1 | 1805 | 309 | 83.10 | 68.29 | 4.62 | 8.23 |
| | | | 2 | 2545 | 1266 | 50.26 | | 14.34 | |
| | | | 3 | 2230 | 635 | 71.52 | | 5.72 | |
| Test Problemleri-I için Genel Ort. = | | | | | | | 59.06 | Genel Ort. = | 6.33 |

Çizelge incelendiğinde, makine sayısı 4, iş sayısı 10, birim iş sayısı 4 olan ilk grup için önerilen TA’nın ortalama 5.13 saniyede, bir saatlik koşumu gerçekleştiren Lingo programından elde edilen sonuçlardan ortalama %32.03 daha iyi bir sonuç verdiği görülmektedir. Test Problemleri-I altında ele alınan ikinci grup için (makine sayısı 4, iş

sayısı 15, birim iş sayısı 6) önerilen TA ortalama 5.62 saniye çalışmış ve Lingo programının bir saatlik çalışmasıyla elde edilen sonuçlardan ortalama %76.87 daha iyi sonuç vermiştir. Son grupta ise amaç fonksiyonu üzerindeki performansın önerilen TA lehinde ortalama %68.29'luk bir farkla gerçekleştiği görülmektedir.

Sonuç olarak; Test Problemleri-I'de yer alan problemler için önerilen TA'nın ortalama 6.33 saniyede, bir saatlik koşumu gerçekleştiren Lingo programından elde edilen sonuçlardan ortalama %59.06 daha iyi bir sonuç verdiği görülmektedir.

Çizelge 4.9. Önerilen Tabu Aramanın Test Problemleri-II üzerindeki performansı.

| MS | IS | BrISS | Örn. No | Amaç Fonksiyonunun Değeri (Z) | | | | TA koşum zamanı (CPU sn.) | Ort. TA koşum Zamanı (CPU sn.) |
|---------------------------------------|----|-------|---------|-------------------------------|------|--------|---------------|---------------------------|--------------------------------|
| | | | | Lingo | TA | % Fark | % Fark (Ort.) | | |
| 8 | 20 | 8 | 1 | 6325 | 1579 | 75.03 | 82.23 | 20.10 | 12.98 |
| | | | 2 | 5324 | 768 | 85.57 | | 9.01 | |
| | | | 3 | 5754 | 800 | 86.10 | | 9.83 | |
| 12 | 20 | 8 | 1 | 5710 | 1429 | 74.97 | 74.79 | 12.03 | 16.84 |
| | | | 2 | 7440 | 1780 | 76.07 | | 27.19 | |
| | | | 3 | 4051 | 1081 | 73.32 | | 11.31 | |
| 12 | 25 | 10 | 1 | 12062 | 3389 | 71.90 | 77.04 | 31.42 | 40.15 |
| | | | 2 | 8217 | 2066 | 74.86 | | 23.62 | |
| | | | 3 | 8950 | 1401 | 84.35 | | 65.42 | |
| Test Problemleri-II için Genel Ort. = | | | | | | | 78.02 | Genel Ort. = | 23.32 |

Çizelge 4.9'a göre, Test Problemleri-II altında ele alınan problemlere önerilen TA'nın çözüm bulması, sırasıyla ortalama 12.98, 16.84 ve 40.15 saniye sürmüştür. Bu süreler zarfında önerilen TA'nın elde ettiği sonuçların, üç saatlik koşumu gerçekleştirilen Lingo programının elde ettiği sonuçlarından ilk problem için ortalama %82.23, ikinci problem için ortalama %74.79 ve son problem grubu için ise ortalama %77.04 daha iyi sonuçlar verdiği görülmektedir.

Sonuç olarak; Test Problemleri-II'de yer alan problemler için önerilen TA'nın ortalama 23.32 saniyede, üç saatlik koşumu gerçekleştiren Lingo programından elde edilen sonuçlardan ortalama %78.02 daha iyi bir sonuç verdiği görülmektedir

Önerilen Tabu Arama, Test Problemleri-I için bir saatlik, Test Problemleri-II için üç saatlik Lingo koşumlarına göre 4.01-65.42 saniye arasında değişen sürelerde %12.24 ile %89.04 arasında daha iyi sonuçlar vermiştir. Amaç değerleri arasındaki fark, problem büyüklüğü arttıkça daha açık bir şekilde görülmektedir. Tüm sonuçlar dikkate alındığında, önerilen Tabu Aramanın problem büyüklüğü arttıkça Lingo'ya göre kısa sürede çok daha başarılı çözümler elde ettiği söylenebilir.

BÖLÜM 5

SONUÇ VE ÖNERİLER

Çalışmada; işlerin bölünebilir özellikte olduğu, özdeş paralel makine çizelgeleme problemi ele alınmıştır. Literatürde yer alan bir probleme bilindiği kadarıyla önceden değinilmemiş meta sezgisel bir yaklaşım olan Tabu Arama ile en iyiye yakın çözümler bulunmaya çalışılmıştır. Bunun için önerilen Tabu Arama elemanları, problemin yapısına uygun olacak şekilde tanımlanmış, sonrasında da Tabu Aramaya has parametre değerleri deney tasarımı yardımıyla belirlenmeye çalışılmıştır.

Önerilen Tabu Araması'nın performansını görmek adına, en iyi sonuçlarla karşılaştırılma yoluna gidilmiştir. Probleme ait en iyi sonuçları üreten bir algoritma var olmadığından, en iyi sonuçlar problem için oluşturulan karma tamsayılı matematiksel modelin Lingo programında koşturulmasıyla elde edilmeye çalışılmıştır. Matematiksel modelle Lingo programında pratik büyüklükteki problemler için en iyi sonuca ulaşmak zaman aldığından, ilk test problemleri küçük boyutta ele alınmıştır. Elde edilen sonuçlar, bu test problemleri içerisindeki hemen hemen her problemde önerilen Tabu yaklaşımının en iyi sonuçları yakaladığını göstermiştir.

Önerilen Tabu Araması'nın daha büyük boyuttaki problemler üzerindeki performansını görmek için, farklı büyüklükteki problemler üzerinde çözümler elde edilmeye çalışılmıştır. Test Problemleri-I için yapılan Lingo koşullarında her bir problemin makul bir zaman diliminde en iyi çözüme ulaşamama durumuna karşı, programa bir saatlik bir süre tanınmıştır. İlgili test problemleri için önerilen Tabu Araması, ortalama 6.33 saniyede zarfında, bir saatlik koşumu gerçekleştiren Lingo programının çözümlerinden ortalama %59.06 daha iyi çözümler üretmiştir. Daha büyük boyutta olan Test Problemleri-II için ise Lingo programının bir saatlik süre zarfında çözüm üretememesi nedeniyle ilgili süre, üç saate çıkarılmıştır. İlgili test problemleri için önerilen Tabu Aramanın, ortalama 23.32 saniyelik bir çalışmayla, üç saatlik

koşumu gerçekleştiren Lingo programının ürettiği çözümlerden ortalama %78.02 daha iyi çözümler ürettiği görülmüştür.

Çalışmada önerilen Tabu Araması yaklaşımıyla kısa sürede, teslim zamanını içeren amaç ele alındığında iyi çözümler üretilerek, teslim zamanlarının önemli bir ölçüt olduğu düşünülen günümüzün müşteri odaklı pazarında rekabet ortamı sağlanabilir. Önerilen Tabu Araması problemin çıkış noktası olan gerçek PCB üretim sistemlerinde uygulama alanı bulabilir.

Bundan sonraki çalışmalarda Tabu Aramanın uzun dönem hafızası da işin içine katılarak çözüm kaliteleri karşılaştırılabilir. Benzer ve farklı paralel makine yapılarını içerecek şekilde problem ele alınabilir. Ek olarak; önerilen Tabu Aramadan daha etkili sonuçlar üretebilecek metotlar geliştirilebilir.

KAYNAKLAR DİZİNİ

- Abdekhodae, A. H., and Wirth, A., 2002, Scheduling parallel machines with a single server: Some solvable cases and heuristics, *Computers and Operations Research*, 29, 295-315.
- Abdekhodae, A. H., and Wirth, A., and Gan, H. S., 2004, Equal processing and equal setup time cases of scheduling parallel machines with a single server, *Computers and Operations Research*, 31, 1867-1889.
- Abdekhodae, A. H., and Wirth, A., and Gan, H. S., 2006, Scheduling two parallel machines with a single server: The general case, *Computers and Operations Research*, 33, 994-1009.
- Akkiraju, R., Keskinocak, P., Murthy, S., and Wu, F., 2001, An agent-based approach for scheduling multiple machines, *Applied Intelligence*, 14, 135-144.
- Allahverdi, A., Gupta, J. N. D., and Aldowaisan, T., 1999, A review of scheduling research involving setup considerations, *Omega*, 27, 219-239.
- Allahverdi, A., Ng, C. T., Cheng, T. C. E., and Kovalyov, M., 2008, A survey of scheduling problems with setup times or costs, *European Journal Of Operational Research*, 187, 985-1032.
- Alpay, Ş., 2003, Kaçırılan teslim zamanı performansı için dinamik stokastik çok makineli atölye tipi üretim ortamı çizelgeleme, *Doktora Tezi, Osmangazi Ü. Fen Bil. Ens.*, 104 sayfa.
- Anglani, A., Grieco, A., Guerriero, E., and Musmanno, R., 2005, Robust scheduling of parallel machines with sequence-dependent setup costs, *European Journal of Operational Research*, 161, 704-720.

KAYNAKLAR DİZİNİ (devam)

- Arnaout, J.P., Rabadi, G., and Mun, J.H., 2006, A dynamic heuristic for the stochastic unrelated parallel machine scheduling problem, *International Journal of Operations Research*, 3, 136-143.
- Azizoğlu, M., 1994, *Paralel Machine Scheduling To Minimize Total Cost Functions*, PhD Thesis, Middle East Technical University, 183 p.
- Azizoglu, M., and Webster, S., 2003, Scheduling parallel machines to minimize weighted flow time with family set-up times, *International Journal of Production Research*, 41, 1199-1215.
- Baker, K., 1994, *Elements of sequencing and scheduling*, Amos Tuck School of Business, D. Hanover, NH.
- Balakrishnan, N., Kanet, J. J., and Sridharan, S. V., 1999, Early/ tardy scheduling with sequence dependent setups on uniform parallel machines, *Computers and Operations Research*, 26, 127-141.
- Bilge, U., Kirac, F., Kurtulan, M., and Pekgun, P.A., 2004, Tabu search algorithm for parallel machine total tardines problem, *Computers and Operations Research*, 31, 397-414.
- Bilgen, B., 1999, *Application of Tabu search to single machine scheduling problems*, Master Thesis, Graduate School of Natural and Applied Sciences of Dokuz Eylül University, 106 p.
- Bitran, G.R., and Gilbert, S.M., 1990, Sequencing production on parallel machines with two magnitudes of sequence-dependent setup cost, *Journal of Manufacturing and Operations Management*, 3, 24-52.

KAYNAKLAR DİZİNİ (devam)

- Blazewicz, J., and Kovalyov, M. Y., 2002, The complexity of two group scheduling problems, *Journal of Scheduling*, 5, 477-485.
- Brucker, P., Kovalyov, M. Y., Shafransky, Y. M., and Werner, F., 1998, Batch scheduling with deadlines on parallel machines, *Annals of Operations Research*, 83, 23-40.
- Brucker, P., Dhaenens-Flippo, C., Knust, S., Kravchenko, S. A. and Werner, F., 2002, Complexity results for parallel machine problems with a single server, *Journal of Scheduling*, 5, 429-457.
- Bruno, J., and Sethi, R., 1978, Tasks sequencing in a batch environment with setup times, *Founding of Control Engineering*, 3, 105-117.
- Chen, B., 1993, A better heuristic for preemptive parallel machine scheduling with batch setup times, *SIAM Journal on Computing*, 22, 1303-1318.
- Chen, J. F., and Wu, T. H., 2006, Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints, *Omega*, 34, 81-89.
- Chen, Z. L., and Powell, W. B., 2003, Exact algorithms for scheduling multiple families of jobs on parallel machines, *Naval Research Logistics*, 50, 823-840.
- Cheng, T.C.E., and Chen, Z-L., 1994, Parallel machine scheduling with batch setup times, *Oper. Res.*, 42, 1171-1174.
- Cheng, T. C. E., and Kovalyov, M. Y., 2000, Parallel machine batching and scheduling with deadlines, *Journal of Scheduling*, 3, 109-123.

KAYNAKLAR DİZİNİ (devam)

- Crauwels, H. A. J., Beullens, P., and Van Oudheusden, D., 2006, Parallel machine scheduling by family batching with sequence-dependent set-up times, *International Journal of Operations Research*, 3, 144-154.
- Dammeyer, F., and Vob, S., 1993, Dynamic tabu list management using reverse elimination method, *Annals of Operations Research*, 41, 31-46.
- Deane R.H., and White, R.H., 1975, Balancing workloads and minimizing setup costs in the parallel processing shop, *Oper Res Q*, 26, 45-53.
- Dearing, P.M., and Henderson, R.A., 1984, Assigning looms in a textile weaving operation with changeover limitations, *Prod. Inv. Manage.*, 25, 23-31.
- Dietrich, B. L., 1989, A two-phase heuristic for scheduling on parallel unrelated machines with setups, IBM TJ Watson Research Center.
- Dunstall, S., and Wirth, A., 2005a, A comparison of branch-and-bound algorithms for a family scheduling problem with identical parallel machines, *European Journal of Operational Research*, 167, 283-296.
- Dunstall, S., and Wirth, A., 2005b, Heuristic methods for the identical parallel machine flowtime problem with set-up times, *Computers and Operations Research*, 32, 2479-2491.
- Dupuy, M., Lamothe, J., Gaborit, P., and Dupont, L., 2005, Efficient neighbors in simulated annealing algorithm to optimize the lead time in a parallel multipurpose machine scheduling problem with setup and calendar constraints, In: *Proceedings of the International Conference on Industrial Engineering and Systems Management*, Marrakech, Morocco, May 16-19, pp. 22-31.

KAYNAKLAR DİZİNİ (devam)

- Elmaghraby, S.E., Guinet, A., and Schellenberger, K.W., 1993, Sequencing on parallel processors: an alternate approach, OR Technical Report No. 273.
- Eom, D.H., Shin, H.J., Kwun, I.H., Shim, J.K., and Kim, S.S., 2002, Scheduling jobs on parallel machines with sequence-dependent family set-up times, *International Journal of Advanced Manufacturing Technology*, 19, 926-932.
- Feng, G., and Lau, H.C., 2005, Efficient algorithms for machine scheduling problems with earliness and tardiness penalties, In: *Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications*, July 18-21, 196-211.
- Franca P.M., Gendreau, M., Laporte G., and Muller, F.M., 1996, A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times, *International Journal of Production Economics*, 43, 79-89.
- Fowler, J.W., Horng, S.M., and Cochran, J.K., 2003, A hybridized genetic algorithm to solve parallel machine scheduling problem machine scheduling problems with sequence dependent setups, *International Journal of Industrial Engineering: Theory Applications and Practice*, 10, 232-243.
- Gambosi, G., and Nicosia, G., 2000, On-line scheduling with setup costs, *Information Processing Letters*, 73, 61-68.
- Gendreau, M., Laporte, G., and Guimaraes, E. M., 2001, A divide and merge heuristic for the multiprocessor scheduling problem with sequence dependent setup times, *European Journal of Operational Research*, 133, 183-189.

KAYNAKLAR DİZİNİ (devam)

- Geoffrion, A.M., and Graves, G.W., 1976, Scheduling parallel production lines with changeover costs: Practical application of a quadratic assignment/LP approach, *Oper Res*, 24, 595-610.
- Gerodimos, A.E., Glass, C.A., Potts, C.N., 2000, Scheduling the production of two-component jobs on a single machine, *European Journal of Operational Research*, 120, 250-259.
- Ghosh, J. B., 1994, Batch scheduling to minimize total completion time, *Operations Research Letters*, 16, 271-275.
- Glass, C. A., Shafransky, Y. M., and Strusevich, V.A., 2000, Scheduling for parallel dedicated machines with a single server, *Naval Research Logistics*, 47, 304-328.
- Glover, F., 1989, Tabu search – part 1, *ORSA Journal on Computing*, 1, 190-206.
- Glover, F., 1990, Tabu search – part 2, *ORSA Journal on Computing*, 2, 4-32.
- Glover, F., and Laguna, M., 1997, *Tabu search*, London:Kluwer Academic Publishers.
- Glover, F., Taillard, E., and Werra, D., 1993, A user's guide to Tabu Search, *Annals of Operations Research*, 41, 3-28.
- Guinet, A., 1990, Textile production systems: a succession of non-identical parallel processor shops, *Journal of the Operational Research Society*, 42, 655-671.
- Guinet, A., and Dussauchhoy, A., 1993, Scheduling sequence dependent jobs on identical parallel machines to minimize completion time criteria, *Int. J. Prod. Res.*, 31, 1579-1594.

KAYNAKLAR DİZİNİ (devam)

- Guirchoun, S., Souhkal, A., and Martineau, P., 2005, Complexity results for parallel machine scheduling problems with a single server in computer systems, In: Proceedings of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications, New York, USA, July 18-21, pp. 232-236.
- Hall, N. G., Potts, C. N., and Sriskandarajah, C., 2000, Parallel machine scheduling with a common server, *Discrete Applied Mathematics*, 102, 223-243.
- Heady, R. B., and Zhu, Z., 1998, Minimizing the sum of job earliness and tardiness in a multi machine system, *International Journal of Production Research*, 36, 1619-1632.
- Hiraishi, K., Levner, E., and Vlach, M., 2002, Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs, *Computers and Operations Research*, 29, 841-848.
- Hurink, J., and Knust, S., 2001, List scheduling in a parallel machine environment with precedence constraints and setup times, *Operations Research Letters*, 29, 231-239.
- Jeong, B., Kim, S.W., and Lee, Y.J., 2001, An assembly scheduler for TFT LCD manufacturing, *Computers and Industrial Engineering*, 41, 37-58.
- Johnson, S. M., 1954, Optimal two and three stage production schedules with set-up times included, *Naval Research Logistic*, 1, 61-68.
- Kim, D.W., Kim, K.H., Jang, W., and Chen, F.F., 2002, Unrelated parallel machine scheduling with setup times using simulated annealing, *Robotics and Computer-Integrated Manufacturing*, 18, 223-231.

KAYNAKLAR DİZİNİ (devam)

- Kim, D.W., Na, D.G., Chen, F.F., 2003, Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective, *Robotics and Computer-Integrated Manufacturing*, 19, 173-181.
- Kim, S.S., and Shin, H.J., 2003, Scheduling jobs on parallel machines: A restricted tabu search approach, *International Journal of Advanced Manufacturing Technology*, 22, 278-287.
- Kim, S.S., Shin, H.J., Eom, D.H., and Kim, C.O., 2003, A due date density-based categorizing heuristic for parallel machines scheduling, *International Journal of Advanced Manufacturing Technology*, 22, 753-760.
- Kim Y. D., Shim S. O., Kim S. B., Choi Y. C., Yoon H. M., 2004, Paralel machine scheduling considering a job splitting property. *International Journal of Production Research*, 42, 4531-4546.
- Koulamas, C. P., 1996, Scheduling two parallel semiautomatic machines to minimize machine interference, *Computers and Operations Research*, 23, 945-956.
- Kravchenko, S. A., and Werner, F., 1997, Parallel machine scheduling problems with a single server, *Mathematical and Computer Modelling*, 26, 1-11.
- Kravchenko, S. A., and Werner, F., 1998, Scheduling on parallel machines with a single and multiple servers, *Otto-von-Guericke-Universitat Magdeburg*, Preprint 30/98, pp. 1-18.
- Kravchenko, S. A., and Werner, F., 2001, A heuristic algorithm for minimizing mean flow time with unit setups, *Information Processing Letters*, 79, 291-296.

KAYNAKLAR DİZİNİ (devam)

- Kruz, M. E., and Askin, R. G., 2001, Heuristic scheduling of parallel machines with sequence-dependent set-up times, *International Journal of Production Research*, 39, 3747-3769.
- Laguna, M., Barnes, W., and Glover, F., 1991, Tabu search methods for a single machine scheduling problem, *Journal of Intelligent Manufacturing*, 2, 63-74.
- Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A.H.G., and Shmoys D. B., 1993, Sequencing and scheduling: algorithms and complexity, In *Logistics of Production and Inventory*, S.C. Graves, A.H.G. Rinnooy Kan, P.H. Zipkin (Eds.), Amsterdam, Netherlands: North Holland, 445-522.
- Lee, Y.H., and Pinedo, M., 1997, Scheduling jobs on parallel machines with sequence-dependent setup times, *European Journal of Operational Research*, 100, 464-474.
- Lee, Y.H., Bhaskaran, K., Pinedo, M., 1997, A heuristic to minimize the total weighted tardiness with sequence dependent setups, *IIE Transactions*, 29, 45-52.
- Leung, J. Y.-T., Ng, C. T., and Cheng, T. C. E., 2008, Minimizing sum of completion times for batch scheduling of jobs with deteriorating processing times, *European Journal of Operational Research*, 187, 1090-1099.
- Liaee, M. M., and Emmons, H., 1997, Scheduling families of jobs with setup times, *International Journal of Production Economics*, 51, 165-176.
- Lin, B. M. T., and Jeng, A. A. K., 2004, Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs, *International Journal of Production Economics*, 91, 121-134.

KAYNAKLAR DİZİNİ (devam)

- Liu, Z., Yu, W., and Cheng, T. C. E., 1999, Scheduling groups of unit length jobs on two identical parallel machines, *Information Processing Letters*, 69, 275-281.
- Marsh, J.D., and Montgomery, D.C., 1973, Optimal procedures for scheduling jobs with sequence-dependent changeover times on parallel processors, *AIIE Technical Papers*, 279-286.
- Mendes, A.S., Muller, F.M., França, P.M., and Moscato, P., 2002, Comparing meta-heuristic approaches for parallel machine scheduling problems, *Production Planning and Control*, 13, 143-154.
- Monma C. L., and Potts, C.N., 1989, On the complexity of scheduling with batch setups, *Oper. Res.*, 37, 798-804.
- Monma C. L., and Potts, C.N., 1993, Analysis of heuristics for preemptive parallel machine scheduling with batch setup times, *Oper. Res.*, 41, 981-993.
- Morton, T.E., Pentico, D.W., 1993, *Heuristic scheduling systems: with applications to production systems and project management*, New York: John Wiley & Sons, 695 p.
- Nessah, F., Yalaoui, F., and Chu, C., 2005, New heuristics for identical parallel machine scheduling with sequence dependent setup times and dates, In: *Proceedings of the International Conference on Industrial Engineering and Systems Management*, Marrakech, Morocco, May 16-19, pp 32-41.
- Ovacik, I.M., and Uzsoy, R., 1993, Worst-case error bounds for parallel machine scheduling problems with bounded sequence-dependent setup times, *Operations Research Letters*, 14, 251-256.

KAYNAKLAR DİZİNİ (devam)

- Ovacik, I.M., and Uzsoy, R., 1995, Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times, *International Journal of Production Research*, 33, 3173-3192.
- Özkazanç, Ü.A., 1999, Atelye tipi üretim ortamında işlerin çizelgelenmesi için yapay sinirsel ağ yaklaşımı, Yüksek lisans Tezi, Osmangazi Üniversitesi Fen Bilimleri Enstitüsü, 104 s.
- Park, Y., Kim, S., and Lee, Y. H., 2000, Scheduling jobs on parallel machines applying neural network and heuristic rules, *Computers and Industrial Engineering*, 38, 189-202.
- Parker, R.G., Deane, R.H., and Holmes, R.A., 1977, On the use of a vehicle routing algorithm for the parallel processor problems with sequence dependent changeover costs, *AIIE Trans*, 9, 155-160.
- Pinedo, M., 1995, *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, 378 p.
- Pinedo, M., 2002, *Scheduling: Theory, Algorithms, and Systems*, 2nd Ed., Prentice Hall, 586 p.
- Radhakrishnan, S., and Ventura, J. A., 2000, Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times, *International Journal of Production Research*, 38, 2233-2252.
- Sabuncuoğlu, İ., 1998, Scheduling with neural networks: a review of the literature and new research directions, *Production Planning and Control*, 9, 2-12.
- Schonberger, R. J., 1988, *Operations Management: Serving the customer*, Business Pub. Inc.

KAYNAKLAR DİZİNİ (devam)

- Schutten, J.M.J., and Leussink R.A.M., 1996, Parallel machine scheduling with release dates, due dates and family setup times, *International Journal of Production Economics*, 46-47:119-125.
- Schuurman, P., and Woeginger, G. J., 1999, Preemptive scheduling with job-dependent setup times, In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 759-767.
- Seçme, G., 2006, Akış tipi çizelgeleme problemlerinin yapay sinir ağları ile modellenmesi, Yüksek Lisans Tezi, Erciyes Üniversitesi Sosyal Bil. Ens., 120 s.
- Serafini P., 1996, Scheduling jobs on several machines with job splitting property, *Operations Research* , 44, 617-628.
- Shim, S. O., and Kim Y. D., 2008, A branch and bound algorithm for an identical paralel machine scheduling problem with a job splitting property, *Computers and Operations Research*, 35, 863-875.
- Silver, E., Pyke, D., and Peterson, R., 1998, *Inventory management and production planning and scheduling*, Wiley & Sons, New York, 754 p.
- Sipper and Bulfin, 1997, *Production: Planning, Control and Integration*, Mcgraw-Hill College, 640 p.
- Sivrikaya-Serifoglu, F., and Ulusoy, G., 1999, Parallel machine scheduling with earliness and tardiness penalties, *Computers and Operations Research*, 26, 773-787.
- So, K. C., 1990, Some heuristics for scheduling jobs on parallel machines with setups, *Management Science*, 36, 467-475.

KAYNAKLAR DİZİNİ (devam)

- Sumichrast, R., and Baker, J.R., 1987, Scheduling parallel processors: an integer linear programming based heuristic for minimizing setup time, *International Journal of Production Research*, 25, 761-771.
- Tahar, D.N., Yalaoui, F., Chu, C., and Amodeo, L., 2006, A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times, *International Journal of Production Economics*, 99, 63-73.
- Tamimi, S. A., and Rajan V. N., 1997, Reduction of total weighted tardiness on uniform machines with sequence dependent setups, In: *Industrial Engineering Research – Conference Proceedings*, pp. 181-185.
- Tang, C. S., and Wittrock, R. J., 1985, Parallel machine scheduling with major and minor setups, IBM TJ Watson Research Center.
- Tang, C. S., 1990, Scheduling batches on parallel machines with major and minor setups, *European Journal of Operational Research*, 46, 28-37.
- Tersine, R., 1985, *Production/Operations Management: Concepts, Structure and Analysis*, Elsevier Science Publishing co., New York.
- Vignier, A., Sonntag, B., and Portmann, M. C., 1999, Hybrid method for a parallel-machine scheduling problem, *IEEE Symposium on Emerging Technologies and Factory Automation, ETFA 1*, 671-678.
- Vollmann, T.E., Berry, W.L., Whybark, O.C., 1997, *Manufacturing Planning And Control Systems*, 4th Ed., BOSTON: IRWIN McGraw-Hill, 836 p.

KAYNAKLAR DİZİNİ (devam)

- Wang, G., and Cheng, T. C. E., 2001, An approximation algorithm for parallel machine scheduling with a common server, *Journal of the Operational Research Society*, 52, 234-237.
- Webster, S., and Azizoglu, M., 2001, Dynamic programming algorithms for scheduling parallel machines with family setup times, *Computers and Operations Research*, 28, 127-137.
- Weng, M. X., Lu, J., and Ren, H., 2001, Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective, *International Journal of Production Economics*, 70, 215-226.
- Wilson, A. D., King, R. E., and Hodgson, T. J., 2004, Scheduling non-similar groups on a flow line: Multiple group setups, *Robotics and Computer-Integrated Manufacturing*, 20, 505-515.
- Wittrock, R. J., 1986, Scheduling parallel machines with setups, IBM TJ Watson Research Center.
- Wittrock, R. J., 1990, Scheduling parallel machines with major and minor setup times, *International Journal of Flexible Manufacturing Systems*, 2, 329-341.
- Xing W., and Zhang J., 2000, Parallel machine scheduling with splitting jobs, *Discrete Applied Mathematics*, 103, 259-269.
- Yalaoui, F., and Chu, C., 2003, An efficient heuristic approach for parallel machine scheduling with job splitting and sequence-dependent setup times, *IIE Transactions*, 35, 183-190.

KAYNAKLAR DİZİNİ (devam)

- Yang, W. H., 2004, Scheduling two-component products on parallel machines, *Omega*, 32, 353-359.
- Yi, Y., and Wang, D. W., 2001a, Tabu search for scheduling grouped jobs on parallel machines, *Journal of Northeastern University*, 22, 188-191.
- Yi, Y., and Wang, D. W., 2001b, Scheduling grouped jobs on parallel machines with setups, *Computer Integrated Manufacturing Systems*, 7, 7-11.
- Yi, Y., and Wang, D. W., 2003, Soft computing for scheduling with batch setup times and earliness-tardiness penalties on parallel machines, *Journal of Intelligent Manufacturing*, 14, 311-322.
- Yi, Y., Chang, H. Y., Wang, J., and Bai, J. C., 2004, Soft computing for parallel scheduling with setup times, *Proceedings of International Conference on Machine Learning and Cybernetics*, 2041-2046.
- Zhu, Z., and Heady, R. B., 2000, Minimizing the sum of earliness/tardiness in multi-machine scheduling: A mixed integer programming approach, *Computers and Industrial Engineering*, 38, 297-305.

EKLER

| | |
|--|----|
| EK.1. Matematiksel Modelin Lingo Programında Yazımı..... | 84 |
| EK.2. Test Problemlerinin Elde Edilmesini İçeren TurboPascal Kodları | 85 |
| EK.3. Başlangıç Çözümün Elde Edilmesini İçeren TurboPascal Kodları | 87 |
| EK.4. Tabu Listelerinin Oluşturulmasını İçeren TurboPascal Kodları | 89 |
| EK.5. Bir Hareketin Tabu Olup Olmadığının Kontrol Edilmesini İçeren TurboPascal Kodları | 91 |
| EK.6. Küçük Test Problemleri İçin Önerilen Tabu Arama'nın ve Lingo Programının Verdiği Sonuçlar | 93 |

EK-1. Matematiksel Modelin Lingo Programında Yazımı

```

SETS:
    Isler: G, BrIsS, HazS, TesZ;
    Makineler;
    MakVePoz (Makineler, Isler);
    IsveMak (Isler, Makineler);
    IsvePoz (Isler, Isler);
    Atama (Isler, Makineler, Isler): x, y, Trd;
ENDSETS

MIN = @SUM(Isler: G);
!1.kısıt;
@FOR(MakVePoz(i,k): @SUM(Isler(j): x(j,i,k)) <= 1);
!2.kısıt;
@FOR(IsveMak(j,i): @SUM(Isler(k): x(j,i,k)) <= 1);
!3.kısıt;
@FOR(Atama(j,i,k): y(j,i,k) <= u*x(j,i,k));
@FOR(Atama(j,i,k): x(j,i,k) <= y(j,i,k));
!4.kısıt;
@FOR(Isler(j): @SUM(MakVePoz(i,k): y(j,i,k)) = u);
!5.kısıt;
@FOR(Atama(j,i,k): Trd(j,i,k) >= @SUM(IsvePoz(l,s) | s #LE# k:
BrIsS(l)*y(l,i,s)+HazS(l)*x(l,i,s))- TesZ(j) - M*(1-x(j,i,k)));
!6.kısıt;
@FOR(Atama(j,i,k): Trd >= 0);
!7.kısıt;
@FOR(Atama(j,i,k): G(j) >= Trd(j,i,k));
!8.kısıt;
@FOR(Atama(j,i,k): @BIN(x));
!9.kısıt;
@FOR(Atama(j,i,k): @GIN(y));
!10.kısıt;
@FOR(Isler: G >= 0);

DATA:
    u=@OLE('C:\Documents and
Settings\CENK\Desktop\Cenk\Verilerim\Verilerim.xls', 'BrIsSay');
    M=@OLE('C:\Documents and
Settings\CENK\Desktop\Cenk\Verilerim\Verilerim.xls', 'BuyukM');
    Makineler, Isler, BrIsS, HazS, TesZ=@OLE('C:\Documents and
Settings\CENK\Desktop\Cenk\Verilerim\Verilerim.xls', 'MakNo', 'IsNo', 'Br
IsSur', 'HazSur', 'TesZam');
ENDDATA

END

```

EK-2. Test Problemlerinin Elde Edilmesini İçeren TurboPascal Kodları.

```

PROGRAM ProblemSeti;
Uses wincrt;
Const
  Nmax = 25;
type
  isVeri=Record
    isNo :byte;
    brisSur:byte;
    isHaz :byte;
    isTes :word
  end;
VAR
  veriler          :array[1..Nmax] of isVeri;
procedure VeriOlustur;
var
  HazUz,AlfaBeta,ilkDeg,sonDeg,i,Toplam:word;
  n,m,u          :byte;
  alfa,beta,TesDagA,TesDagB      :real;
  txtVeri        :text;
begin
  randomize;
  Write('Makine Sayısını Giriniz: ');Readln(m);
  Write('İş Sayısını Giriniz [Maksimum 25]: ');Readln(n);
  Write('Birim İş Sayısını Giriniz: ');Readln(u);
  Writeln('Hazirlik Suresinin Uzunlugu:');writeln(' *Kisa:U~[5,60]* = 1,
*Orta:U~[60,120]* = 2, *Uzun:U~[120,180]* = 3');
  Readln(HazUz);
  Writeln('Alfa ve Beta Degerleri:');writeln(' [(0,0.4) = 1], [(0,0.8) = 2], [(0,1.2) = 3],
[(0.5,1.0) = 4]');
  Readln(AlfaBeta);
  case HazUz of
    1: begin
      ilkDeg:=5;
      sonDeg:=60;
    end;
    2: begin
      ilkDeg:=60;
      sonDeg:=120;
    end;
    3: begin
      ilkDeg:=120;
      sonDeg:=180;
    end;
  end;
end;

```

EK.2. (devam) Test Problemlerinin Elde Edilmesini İçeren TurboPascal Kodları

```

4: begin
    ilkDeg:=5;
    sonDeg:=180
end;
end;
case AlfaBeta of
1: begin
    alfa:=0.0;
    beta:=0.4;
end;
2: begin
    alfa:=0.0;
    beta:=0.8;
end;
3: begin
    alfa:=0.0;
    beta:=1.2;
end;
4: begin
    alfa:=0.5;
    beta:=1.0;
end;
end;
for i:=1 to n do begin
    veriler[i].isNo:=i;
    veriler[i].BrisSur:=round(random(55)+5);
    veriler[i].isHaz:=round(random(sonDeg-ilkDeg)+ilkDeg)
end;
Toplam:=0;
for i:=1 to n do Toplam:=Toplam+veriler[i].isHaz+u*veriler[i].brisSur;
TesDagA:=alfa*(Toplam/m); TesDagB:=beta*(Toplam/m);
for i:=1 to n do veriler[i].isTes:=round(random*(TesDagB-TesDagA)+TesDagA);
clrscr;
Assign(txtVeri,'dosVeri.txt');
Rewrite(txtVeri);
for i:=1 to n do begin
writeln(txtVeri,veriler[i].isNo:2,veriler[i].brisSur:7,veriler[i].isHaz:8,veriler[i].isTes:8);
end;
Close(txtVeri);
write('m=');write(m:2);write(', n=');write(n:3);write(', u=');write(u:2);
writeln(' için gerekli veriler oluşturulup, "dosVeri.txt" dosyasına yazıldı.')
end;
BEGIN
    VeriOlustur;
END.

```

EK-3. Başlangıç Çözümün Elde Edilmesini İçeren Turbo Pascal Kodları

```

procedure BasCozEDD;
var
  EDDsira:array[1..n] of isVeri;
  degis :isVeri;
  i,j  :byte;
begin
  for i:=1 to n do EDDsira[i]:=veriler[i];
  for i:=1 to n-1 do begin
    for j:=i+1 to n do begin
      if EDDsira[i].isTes>EDDsira[j].isTes then
        begin
          degis:=EDDsira[i];
          EDDsira[i]:=EDDsira[j];
          EDDsira[j]:=degis
        end;
    end;
  end;
  for i:=1 to m do begin
    for j:=1 to n do begin
      with baslPMC.mak[i].sira[j] do
        begin
          isTipi:=0;
          brisSay:=0
        end;
    end;
  end;
  for i:=1 to m do begin
    for j:=1 to n do
      baslPMC.mak[i].sira[j].isTipi:=EDDsira[j].isNo;
    end;
  end;
end;

```

EK-3. (devam) Başlangıç Çözümün Elde Edilmesini İçeren Turbo Pascal Kodları

```
for i:=1 to m do
  begin
    j:=i;
    repeat
      baslPMC.mak[i].sira[j].brisSay:=u;
      j:=j+m;
    until j>n
  end;
amac(baslPMC);
eniyePMC:=baslPMC;
end;
```

EK-4. Tabu Listelerinin Oluşturulmasını İçeren TurboPascal Kodları

```
procedure TabuListeSEkle(tabu:tabulstSVeri);
```

```
  var
    eleman:byte;
  begin
    if tabugirisS<tabusureS then
      begin
        tabugirisS:=tabugirisS+1;
        tabulisteS[tabugirisS]:=tabu
      end
    else
      begin
        for eleman:=1 to tabusureS-1 do
          tabulisteS[eleman]:=tabulisteS[eleman+1];
          tabulisteS[eleman+1]:=tabu;
          tabugirisS:=tabugirisS+1
        end;
      end;
  end;
```

```
procedure TabuListeIEkle(tabu:tabulstIVeri);
```

```
  var
    eleman:byte;
  begin
    if tabugirisI<tabusureI then
      begin
        tabugirisI:=tabugirisI+1;
        tabulisteI[tabugirisI]:=tabu
      end
    else
      begin
        for eleman:=1 to tabusureI-1 do
```

EK-4. (devam) Tabu Listelerinin Oluşturulmasını İçeren TurboPascal Kodları

```
        tabulisteI[eleman]:=tabulisteI[eleman+1];  
    tabulisteI[eleman+1]:=tabu;  
    tabugirisI:=tabugirisI+1  
end;  
end;
```

EK-5. Bir Hareketin Tabu Olup Olmadığının Kontrol Edilmesini İçeren TurboPascal Kodları

```
function TabudurS(kPMC:pmcVeri;mkns,y1:byte):boolean;
var
  eleman:byte;
begin
  TabudurS:=false;
  if tabugirisS<>0 then
    begin
      for eleman:=1 to tabusureS do
        begin
          if mkns=tabulisteS[eleman].mkno then
            if kPMC.mak[mkns].sira[y1].isTipi=tabulisteS[eleman].is1 then
              if kPMC.mak[mkns].sira[y1].brisSay>=tabulisteS[eleman].bris1 then
                if y1<=tabulisteS[eleman].yer then
                  begin
                    TabudurS:=true;
                    exit
                  end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
```

```
function TabudurI(mkn,isnosu:byte):boolean;
var
  eleman:byte;
begin
  TabudurI:=false;
  if tabugirisI<>0 then
    begin
```


EK-5. (devam) Bir Hareketin Tabu Olup Olmadığının Kontrol Edilmesini İçeren TurboPascal Kodları

```
for eleman:=1 to tabusureI do
  begin
    if isnosu=tabulisteI[eleman].isno then
      if mkn=tabulisteI[eleman].mkno then
        begin
          TabudurI:=true;
          exit
        end;
      end;
    end;
  end;
end;
```

EK-6. Küçük Test Problemleri İçin Önerilen Tabu Aramanın ve Lingo Programının Verdiği Sonuçlar

| Mak Say. | İş Say. | Br. İş Say. | Problem No | Lingo | | TA | |
|----------|---------|-------------|------------|-------|---------------|-----|---------------|
| | | | | Z | Süre (CPU sn) | Z | Süre (CPU sn) |
| 2 | 2 | 5 | 1 | 313 | 0.0 | 313 | 1.1 |
| | | | 2 | 91 | 0.0 | 91 | 0.9 |
| | | | 3 | 144 | 0.0 | 144 | 1.0 |
| | | | 4 | 133 | 0.0 | 133 | 2.0 |
| | | | 5 | 243 | 0.0 | 243 | 2.3 |
| | | | 6 | 144 | 0.0 | 144 | 1.1 |
| | | | 7 | 42 | 0.0 | 42 | 1.1 |
| | | | 8 | 160 | 0.0 | 160 | 1.0 |
| | | | 9 | 99 | 0.0 | 99 | 1.0 |
| | | | 10 | 0 | 0.0 | 0 | 1.0 |
| 2 | 3 | 5 | 1 | 5 | 0.0 | 5 | 1.0 |
| | | | 2 | 17 | 0.0 | 17 | 1.0 |
| | | | 3 | 56 | 0.0 | 56 | 1.0 |
| | | | 4 | 323 | 1.0 | 323 | 2.2 |
| | | | 5 | 113 | 1.0 | 113 | 2.0 |
| | | | 6 | 198 | 0.0 | 198 | 1.1 |
| | | | 7 | 19 | 1.0 | 19 | 1.1 |
| | | | 8 | 67 | 1.0 | 67 | 1.0 |
| | | | 9 | 80 | 1.0 | 80 | 1.1 |
| | | | 10 | 16 | 0.0 | 16 | 0.9 |
| 2 | 4 | 5 | 1 | 286 | 19.0 | 286 | 1.2 |
| | | | 2 | 107 | 62.0 | 107 | 1.1 |
| | | | 3 | 297 | 153.0 | 297 | 1.0 |
| | | | 4 | 112 | 42.0 | 112 | 1.0 |
| | | | 5 | 276 | 74.0 | 276 | 1.0 |
| | | | 6 | 170 | 39.0 | 170 | 1.0 |
| | | | 7 | 5 | 31.0 | 5 | 1.0 |
| | | | 8 | 459 | 139.0 | 459 | 1.0 |
| | | | 9 | 259 | 115.0 | 259 | 1.0 |
| | | | 10 | 163 | 35.0 | 163 | 1.1 |
| 3 | 3 | 5 | 1 | 58 | 83.0 | 58 | 1.1 |
| | | | 2 | 153 | 101.0 | 153 | 1.0 |
| | | | 3 | 161 | 92.0 | 161 | 1.2 |
| | | | 4 | 136 | 45.0 | 136 | 1.1 |
| | | | 5 | 235 | 71.0 | 242 | 1.9 |
| | | | 6 | 144 | 42.0 | 144 | 1.1 |
| | | | 7 | 55 | 41.0 | 55 | 1.0 |
| | | | 8 | 62 | 50.0 | 62 | 1.5 |
| | | | 9 | 67 | 252.0 | 67 | 1.1 |
| | | | 10 | 325 | 64.0 | 325 | 1.1 |
| 3 | 4 | 5 | 1 | 125 | 10702.0 | 125 | 1.9 |
| | | | 2 | 54 | 6104.0 | 54 | 1.0 |
| | | | 3 | 219 | 10269.0 | 219 | 1.2 |
| | | | 4 | 319 | 2858.0 | 319 | 1.5 |
| | | | 5 | 367 | 3727.0 | 367 | 1.1 |
| | | | 6 | 10 | 2974.0 | 10 | 1.1 |
| | | | 7 | 198 | 2440.0 | 198 | 0.9 |
| | | | 8 | 174 | 2667.0 | 174 | 1.3 |
| | | | 9 | 233 | 3375.0 | 233 | 1.0 |
| | | | 10 | 266 | 3596.0 | 266 | 1.0 |