S.Ü. Müh.-Mim. Fak. Derg., c.21, s.1-2, 2006
J. Fac.Eng.Arch. Selcuk Univ., v.21, n.1-2, 2006

PARALEL MAKİNELİ ÇİZELGELEMEDE TOPLAM TAMAMLANMA ZAMANI VE MAKSİMUM GECİKMENİN ENKÜÇÜKLENMESİ

Tamer EREN[1] ve Ertan GÜNER[2]
[1]*Kırıkkale Üniversitesi, Endüstri Mühendisliği Bölümü, 71450 Kırıkkale,*
[2]*Gazi Üniversitesi, Endüstri Mühendisliği Bölümü, 06570 Maltepe, Ankara,*
teren@kku.edu.tr erguner@gazi.edu.tr

Makalenin Geliş Tarihi: 26.12.2005

ÖZET: Bu çalışmada iki ölçütlü özdeş iki paralel makineli çizelgeleme problemi incelenmiştir. Problemin amaç fonksiyonu toplam tamamlanma zamanı ve maksimum gecikmenin ağırlıklı toplamını en küçüklemektir. Tamamlanma zamanı ve maksimum gecikme çizelgeleme literatüründe en çok göz önüne alınan ölçütlerdendir. NP-zor yapıda olan bu problemin çözümü için, $n^3/2+3n^2/2+n/2$ değişkenli ve $3n^2$ kısıtlı bir tamsayılı programlama modeli geliştirilmiştir (burada n iş sayısını ifade etmektedir). Tam sayılı programlama modelinin hesaplama zamanı ve yüksek hesaplama karmaşıklığı dolayısı ile 20 işe kadar olan problemlerin çözümleri gerçekleştirilebilmiştir. Problemin daha büyük boyutlu çözümlerini gerçekleştirmek için çizelgelemede iyi bilinen dağıtım kurallarına göre belirlenen sıralar başlangıç çözümü olarak alınarak tabu arama yöntemleri (Tabu I, Tabu II ve Tabu III) ve rassal arama yöntemi geliştirilmiş ve problemin 1000 işe kadar çözümleri bu yöntemlerle belirlenmiştir.

Anahtar kelimeler: Paralel makineli çizelgeleme, iki ölçüt, tamsayılı programlama, sezgisel yöntemler.

Minimization of Total Completion Time and Maximum Tardiness on Parallel Machine Scheduling

ABSTRACT: In this study bicriteria identical two parallel machine scheduling problem is considered. The objective function of the problem is minimization of the weighted sum of completion time and maximum tardiness. Total completion time and maximum tardiness are widely used performance measures in scheduling literature. An integer programming model with $n^3/2+3n^2/2+n/2$ variables and $3n^2$ constraints (where n is the number of jobs) is developed for the problem which belongs to NP-hard class. Because of the lengthy computing time and high computing complexity of the integer programming model, the problem with up to 20 jobs can be solved. A random search method and tabu search based heuristic algorithms (Tabu I, Tabu II and Tabu III) are presented to solve large size problems. To improve the performance of tabu search algorithms the sequences found from the well known dispatching rules are taken as an initial solution of tabu search algorithms. According to computational results the tabu search algorithm is effective in finding problem solutions with up to 1000 jobs.

Keywords: Identical parallel machine, bicriteria, integer programming, heuristic methods.

GİRİŞ

Çok ölçütlü çizelgeleme, son yıllarda araştırmacıların en çok ilgisini çeken konulardan birisidir. Bu konuda tek makineli ve akış tipi sistemler üzerinde literatürde oldukça çalışma yapılmış olmasına karşın paralel makineli sistemlerde daha az çalışma yapılmıştır (Eren, 2004; Eren ve Güner 2002, 2004). Bu çalışmada iki ölçütlü paralel iki makineli çizelgeleme problemi göz önüne alınmıştır. Dikkate alınan performans ölçütleri çizelgelemede en önemli ölçütlerden olan toplam tamamlanma zamanı ile maksimum gecikme ölçütleridir.

Toplam tamamlanma zamanının en küçüklenmesi, yarı ürün stoklarını düşürmenin bir göstergesi olarak dikkate alınırken, maksimum gecikme ölçütünü en küçükleyerek gecikmeden kaynaklanan cezaları azaltmak mümkün olmaktadır.

Çok ölçütlü paralel makineli sistemlerde maksimum tamamlanma zamanı ve toplam tamamlanma zamanını Eck ve Pinedo (1993), Gupta ve diğ. (2000) ve Gupta ve Ho (2001) iki makineli durumda incelerken, Gupta ve Ruis-Torres (2000) ve Lin ve Liao (2004) ise m makineli durumda çözüm yaklaşımları geliştirmişlerdir. Maksimum tamamlanma zamanı ile maksimum gecikme ölçütlerini ise Mohri ve diğ. (1999) üç makineli durum için, Suresh ve Chaudhuri (1996), ise m makineli durum için incelemişlerdir. Sarin ve Hariharan (2000) ise maksimum gecikme ve geciken iş sayısını enküçükleme problemini ele almışlardır. Yapılan çalışmaların tümünde dal-sınır yöntemi kullanılmıştır. Büyük boyutlu problemler içinde sezgisel yaklaşımlar geliştirmişlerdir.

Bu çalışmada iki ölçütlü paralel özdeş iki makineli çizelgeleme problemi ele alınmış ve amaç fonksiyonu içinde toplam tamamlanma zamanı ve maksimum gecikmenin ağırlıklı toplamını minimize etme seçilmiştir. Problem için tamsayılı programlama modeli geliştirilmiş ve 20 işe kadar optimal çözümler bulunmuştur. Daha büyük boyutlu problemlerin çözümü içinde tabu arama yöntemi kullanılmıştır.

Ele aldığımız iki ölçütlü iki paralel makineli $P2//\alpha\sum C+\beta T_{\max}$ problemi, NP-zor yapıda bir problemdir. Çünkü bu problemin özel hali olan $\alpha=0$ olduğu durumda $P2//T_{\max}$ problemi NP-zor yapıdadır (Lenstra ve diğ. 1977).

Çalışmanın ikinci bölümünde ele alınan iki ölçütlü paralel makineli problem tanımlanmıştır. Üçüncü bölümde ise problemin en iyi çözümlerini bulmak için tamsayılı programlama modeli verilmiş ve verilen model sayısal bir örnek üzerinde gösterilmiştir. Dördüncü bölümde ise büyük boyutlu problemleri çözmek için sezgisel yöntemler önerilmiştir. Deneysel sonuçlar beşinci bölümde verilirken, son bölümde yapılan çalışmanın sonuçları değerlendirilmiş ve gelecekte yapılabilecek çalışmalar hakkında öneriler sunulmuştur.

PROBLEMİN TANIMLANMASI

Atölyeye gelen n iş sıfırıncı zamanda işlem için hazırdır. Paralel makineli sistemde gelen işler ($j=1,2,...,n$) mevcut paralel makinelerin ($i=1,2$) herhangi birinde işlem görebilir. p_j ve d_j, j işinin işlem zamanını ve teslim tarihini göstermektedir. C_j sırasıyla j işinin tamamlanma zamanı ifade etmektedir. Maksimum gecikme ise $T_{\max}=\max_{j=1}^{n}\{C_j-d_j,0\}$ olarak tanımlanmaktadır. Makineler özdeş makinelerdir. Bu makinelere işlerin atanması örneği olarak 10 işli bir durum dikkate alındığında (n_1,n_2)=(9,1);(8,2);(7,3);(6,4) ve (5,5) olmaktadır. Burada n_1 ve n_2 birinci ve ikinci makinelere atanan işlerin sayısını göstermektedir.

Çalışmada kullanılan diğer varsayımlar ise : makine hazırlık zamanları önceden bilinmekte olup işlem zamanına dahil edilmiştir. İş kesintisine izin verilmeyip başlanan bir iş makinede tamamlanmadan başka bir iş başlayamaz ve makinelerin çizelgeleme periyodu süresince sürekli çalıştığı varsayılmaktadır. Ayrıca bir makinede aynı anda tek bir iş yapılabilmektedir.

TAMSAYILI PROGRAMLAMA MODELİ

Ele alınan iki ölçütlü paralel makineli çizelgeleme problemi için geliştirilen tamsayılı programlama modeli, $n^3/2+3n^2/2+n/2$ değişkenli ve $3n^2$ kısıtlıdır (n iş sayısını göstermektedir). Problemde kullanılan parametreler, değişkenler ve model aşağıda açıklanmıştır.

Parametreler

n iş sayısı $j=1,2,...,n$.

m makine sayısı $i=1,2$.

α Toplam tamamlanma zamanı ağırlık değeri $\alpha\geq 0$

β maksimum gecikme ağırlık değeri $\alpha+\beta=1$

p_j j işinin işlem zamanı

$p_j=p_{ji}$ $i=1,2$ $j=1,2,...,n$.

d_j j işinin teslim tarihi $j = 1,2,...,n$.

n_i i. makinedeki iş sayısı $\sum_{i=1}^{2} n_i = n$ $i = 1,2$.

Karar Değişkeni

$$X_{jik} = \begin{cases} 1 & \text{j. iş i. makinede k. sıradak işş atanıtan} \\ 0 & \text{dd} \end{cases}$$
$$i = 1,2 \qquad j,k = 1,2,...,n$$

Yardımcı Değişkenler

$p_{[ki]}$ i. makinada k. sıraya atanan işin işlem zamanı

$$p_{[ki]} = \sum_{j=1}^{n} X_{jik} p_{ji} \quad i = 1,2 \quad k = 1,2,...,n_i \qquad (1)$$

C_{ki} i. makinada k. sıradaki işin tamamlanma zamanı,

$$C_{1i} = p_{[1i]} \quad \text{ve} \quad C_{ki} \geq C_{k-1,i} + p_{[ki]}$$
$$i = 1,2 \qquad k = 2,3,...,n_i \qquad (2)$$

d_{ik}^* i. makinada k. sıraya atanan işin teslim tarihi

$$d_{ik}^* = \sum_{j=1}^{n} X_{jik} d_j \quad i = 1,2 \quad k = 1,2,...,n_i \qquad (3)$$

$T_{\max}$ maksimum gecikme

$$T_{\max} \geq C_{ki} - d_{ik}^* \quad i = 1,2 \quad k = 1,2,...,n_i \qquad (4)$$

Matematiksel Programlama Modeli

Amaç fonksiyonu

$$Min \quad \alpha \sum C + \beta T_{\max}$$

Kısıtlar:

$$\sum_{j=1}^{n} X_{jik} = 1 \qquad i = 1,2 \quad k = 1,2,...,n_i . \qquad (5)$$

$$\sum_{i=1}^{m} \sum_{k=1}^{n_i} X_{jik} = 1 \quad j = 1,2,...,n \qquad (6)$$

$$\sum_{j=1}^{n} X_{jik} = 0 \text{ veya } 1 \qquad j = 1,2,...,n$$
$$i = 1,2 \qquad k = 1,2,...,n_i \qquad (7)$$

(1)-(4) nolu yardımcı değişken kısıtları

Modele ait sayısal bir örnek

Paralel özdeş iki makineli 10 işe ait işlerin işlem zamanları ve teslim tarihleri Tablo 1'de verilmiştir. Toplam tamamlanma zamanı ve maksimum gecikme ölçütlerinin üç farklı ağırlık değerleri, (α,β)=(0.25, 0.75); (0.50, 0.50) ve (0.75, 0.25), için önerilen modelle belirlenen çözümleri aşağıda gösterilmiştir:

Sayısal örnek çözümü

Önerilen modelle problem çözüldüğünde bulunan sonuçları Tablo 2'de verilmiştir. Ayrıca Tablo 2'de Her bir ağırlık değeri α ve β için en iyi sonuçlar koyu renk ile gösterilmiştir (α,β)=(0.25, 0.75) ve (0.50, 0.50) için birinci ve ikinci makineye atanan iş sayısı sırası ile yedi ve üç olmaktadır. (α,β)=(0.75, 0.25) için ise işler makineye eşit olarak dağılmıştır.

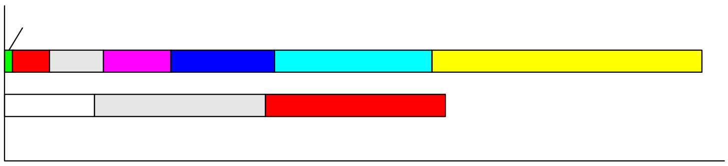
Örnek problemin her bir ağırlık değeri için optimal sonuçların Gantt şemaları Şekil 1'de verilmiştir. Tablo 3'de de örnek problemin çözümünde her bir makineye atanan işleri göstermektedir. En iyi sonuçlar yine koyu renk ile gösterilmiştir.

Tablo 1. Sayısal örnek verileri.
Table 1. *Numerical example data.*

j	1	2	3	4	5	6	7	8	9	10
p_j	3	69	76	41	99	16	80	24	29	45
d_j	119	471	47	330	352	453	259	353	119	455

Tablo 2. Sayısal örnek çözüm sonuçları.
Table 2. *Numerical example solution results.*

	(α,β)		
(n_1,n_2)	(0.25, 0.75)	(0.50, 0.50)	(0.75, 0.25)
(5,5)	302.00	560.50	**799.75**
(6,4)	299.50	560.50	801.25
(7,3)	**270.00**	**540.00**	810.00
(8,2)	320.00	611.00	902.00
(9,1)	388.00	739.00	1081.50



(a) (α,β)=(0.25,0.75)

(b) (α,β)=(0.50,0.50)

(c) (α,β)=(0.75,0.25)

Şekil 1. En iyi sonuçların Gantt Şemaları.
Figure 1. *Gantt diagrams for optimal results.*

Tablo 3. Sayısal örnekteki her bir makineye atanan iş sayıları.
Table 3. *Assigned jobs to each machine in numerical example.*

	(α,β)		
$(n_1;n_2)$	(0.25,0.75)	(0.50,0.50)	(0.75,0.25)
(5;5)	M_1: 1-3-4-10-7	M_1: 6-9-4-10-5	**M_1: 6-9-4-2-5**
	M_2: 6-8-9-2-5	M_2: 1-8-3-2-7	**M_2: 1-8-10-3-7**
(6;4)	M_1: 6-8-9-4-2-5	M_1: 1-8-9-4-10-5	M_1: 1-6-9-10-2-5
	M_2: 1-3-10-7	M_2: 6-3-2-7	M_2: 8-4--3-7
(7;3)	**M_1: 1-6-8-9-10-2-5**	**M_1: 1-6-8-9-4-2-5**	M_1: 1-6-8-9-10-2-5
	M_2: 4-3-7	**M_2: 10-3-7**	M_2: 4-3-7
(8;2)	M_1: 1-6-8-9-4-10-2-5	M_1: 1-6-8-9-4-10-2-5	M_1: 1-6-8-9-4-10-2-5
	M_2: 3-7	M_2: 3-7	M_2: 3-7
(9;1)	M_1: 1-6-8-9-4-10-7-5-2	M_1: 1-6-8-9-4-10-2-7-5	M_1: 1-6-8-9-4-10-2-7-5
	M_2: 3	M_2: 3	M_2: 3

1

M_1 6 8

SEZSİSEL YÖNTEMLER

Önerilen matematiksel programlama modeli ile ancak küçük boyutlu problemler çözülebilmektedir. Halbuki uygulamalarda daha büyük boyutlu problemleri çözmek gerekebilir. Bunun için ele alınan büyük boyutlu problemlerin çözümünde rassal arama ve tabu arama yöntemi kullanılacaktır.

Rassal arama

Belli boyutta bir örnek kümesi seçilir. Bunlar arasında en iyi çözüm referans alınır ve her iterasyonda güncelleştirilir. Rassal aramanın iki parametresi vardır: Örnek büyüklüğü ve iterasyon sayısı. Bir sonraki kısımda anlatılacağı gibi tabu arama yöntemi ile aynı sayıda çözümü incelemek için örnek büyüklüğü (n-1) ve iterasyon sayısı da 3n alınmıştır.

Tabu Arama

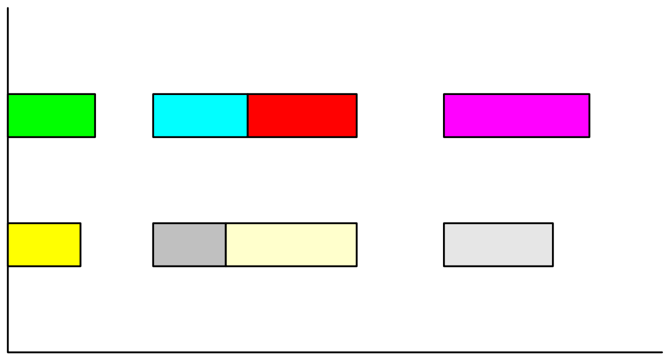
İlk olarak Glover (1986) tarafından ortaya atılan tabu arama yöntemi, bu çalışmada ele alınan problemin çözümünde kullanılan sezgisel yöntemdir. Bu yöntem, en iyi veya en iyiye yakın çözümleri bulmak için çözüm uzayını araştırır. Kombinatoryal problemlerde kullanılan sezgisel en iyileme tekniklerinden en çok kullanılan yöntemlerden biridir. Tabu arama, seçilen herhangi bir başlangıç çözümü ile aramaya başlar. Mevcut çözümün tanımlanan bir hareket mekanizmasına göre komşuluğu oluşturulur ve bu komşuluk içinden en iyi amaç değerine sahip olan çözüm eğer tabu sınıfına girmiyorsa yeni mevcut çözüm olarak seçilir. Yöntemde tabu sınıflarının belirlenmesi için kısa dönemli hafıza (tabu listesi) kullanılır. Belli bir iterasyon seviyesinde veya iyileşme olmadığında arama durdurulur.

Tabu arama yöntemi problem için dört durumda ele alınmıştır.

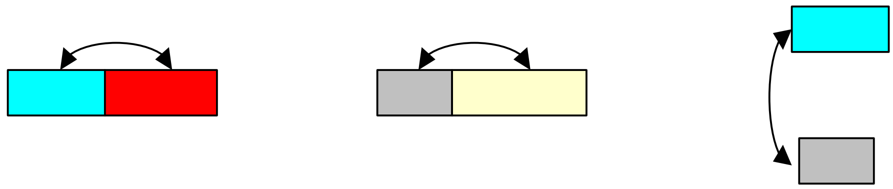
Başlangıç çözümleri: Rassal, SPT (en kısa işlem zamanı) ve EDD (en erken teslim tarihi) kuralları kullanılmıştır.
Komşu arama stratejisi: Komşu arama stratejisi olarak bitişik iş çiftlerinin yer değiştirilmesi (API) kullanılmıştır. API stratejisi 3 durumda aşağıdaki şekilde ele alınmıştır:
API-I: 1. makinedeki p_{j1} işi ile $p_{j+1,1}$ bitişik iş çiftinin yer değiştirmesi.
API-II: 2. makinedeki p_{k2} işi ile $p_{k+1,2}$ bitişik iş çiftinin yer değiştirmesi.
API-III 1. makinedeki p_{j1} işi 2. makinedeki p_{k2} iş çiftinin yer değiştirmesi. Paralel makinenin Gantt şeması Şekil 2'de komşu arama stratejileri ise Şekil 3'de gösterilmiştir.

Tabu listesi uzunluğu: Tabu listesi uzunluğu iş sayısı n'e göre belirlenmiş ve $\sqrt{n}$ 'in tamsayı değeri alınmıştır.

Durdurma kriteri: Problem için 3n iterasyonda tabu arama yöntemi son verilmesi istenmektedir. Tabu aramanın parametreleri toplu olarak Tablo 4'de verilmiştir.



Şekil 2. Paralel makinenin Gantt şeması.
Figure 2. *Gantt diagram of parallel machine.*



(a) (b) (c)

Şekil 3. API komşu arama stratejileri.

***Figure 3**. API neighborhood search strategies.*

(a) API-I: 1. makinedeki bitişik iş çiftinin yer değiştirmesi.

***(a) API-I:** Adjacent pairwise interchange in first machine.*

(b) API-II: 2. makinedeki bitişik iş çiftinin yer değiştirmesi.

***(b) API-II:** Adjacent pairwise interchange in second machine.*

(c) API-III: 1. makinedeki iş ile 2. makinedeki işin yer değiştirmesi.

***(c) API-III:** Interchange the job in first machine with the job in second machine.*

Tablo 4. Tabu arama parametreleri.

***Table 4.** Tabu search parameters.*

Sezgiseller	Tabu-I	Tabu-II	Tabu-III
Başlangıç çözümü	Rassal	SPT	EDD
Tabu listesi uzunluğu	$\sqrt{n}$	$\sqrt{n}$	$\sqrt{n}$
Komşu arama stratejisi	Rassal{API-I,API-II, API-III}	Rassal{API-I,API-II, API-III}	Rassal{API-I,API-II, API-III}
Durdurma kriteri	$3n$ iterasyon	$3n$ iterasyon	$3n$ iterasyon

p_{j1} $p_{j+1,1}$

DENEYSEL SONUÇLAR

Yapılan çalışmada bütün deneysel testler için Pentium IV/2 GHz 512 RAM kapasiteli kişisel bilgisayar kullanılmıştır. Ele alınan problemin en iyi çözümlerini bulmak için Hyper LINDO/PC 6.01 programı kullanılmıştır. Sezgisel yöntemler C++ builder ile kodlanmıştır. İşlem zamanları p_j, 1 ile 100 arasında, teslim tarihleri d_j ise 0 ile $\sum_{j=1}^{n} p_j/2$ arasında düzgün dağılımdan üretilmiştir. Deney seti toplu olarak Tablo 5'de verilmiştir. Tablo 5'de görüldüğü gibi toplam 90 problem çözülmüştür.

Yalnız her bir iş sayısı için alt problemler seti çözülmüştür. Örneğin 10 iş için (n_1, n_2)=(9,1), (8,2), (7,3), (6,4) ve (5,5) olmak üzere 5, 15 iş için (n_1, n_2)=(14,1), (13,2), (12,3), (11,4), (10,5), (9,6) ve (8,7) olmak üzere 7 ve 20 iş için (n_1, n_2)=(19,1), (18,2), (17,3), (16,4), (15,5), (14,6), (13,7), (12,8), (11,9), (10,10) olmak üzere 10 alt problem oluşturup çözülmüştür.

Tablo 5. Problemin deneysel seti.

***Table 5.** Experimental set of problem.*

Parametreler	**Alternatif**	**Değerler**
Ağırlıklar (α,β)	3	(0.25,0.75); (0.50,0.50); (0.75,0.25)
İş sayısı, n	3	10,15,20
İşlem zamanı p_j	1	[1,100]
Teslim tarihi d_j	1	[0, $\sum p_j/2$]
Çözülen problem		10
Toplam problem		3×3×1×1×10=90

Tablo 6'da ele alınan iki ölçütlü paralel makineli problemin tamsayılı programlama çözüm süreleri verilmektedir. Tablo 6'dan da görüleceği gibi makinelere atanan işler ne kadar dengeli dağılıyor ise problemin zorluğu o derece artmakta ve çözüm süresi uzamaktadır. Ayrıca maksimum gecikme ölçütüne ilişkin ağırlık değeri (β) arttıkça problemin çözümü zorlaşmaktadır. Örneğin 10 işli grupta en uzun CPU zamanı α =0.25, β=0.75 için gerçekleşirken (4.01), 15 ve 20 işli gruplar içinde benzer sonuçlar elde edilmiştir (sırası ile 288.45s ve 4166.43s). Ayrıca, tüm ağırlık değerleri için iş sayısı arttıkça CPU zamanlarının oldukça hızla arttığına dikkat edilmelidir ki bu problemin zorluğunun bir göstergesidir. Tablo 6 'ye ek olarak Şekil 4'de her bir iş grubu için çözüm süresi ağırlık değerlerine göre gösterilmiştir.

Problem sezgisel yöntemlerle çözüldüğünde, sezgisellerin çözüm hataları Tablo 7'de verilmiştir. Sezgisellerin ağırlıklara göre ortalama hata değerleri de Şekil 5'de verilmiştir. Hata değerleri (8) nolu ifade ile hesaplanmıştır.

$$\text{hata} = \frac{\text{sezgisel çözüm degeri - optimal çözüm degeri}}{\text{optimal çözüm degeri}} \quad (8)$$

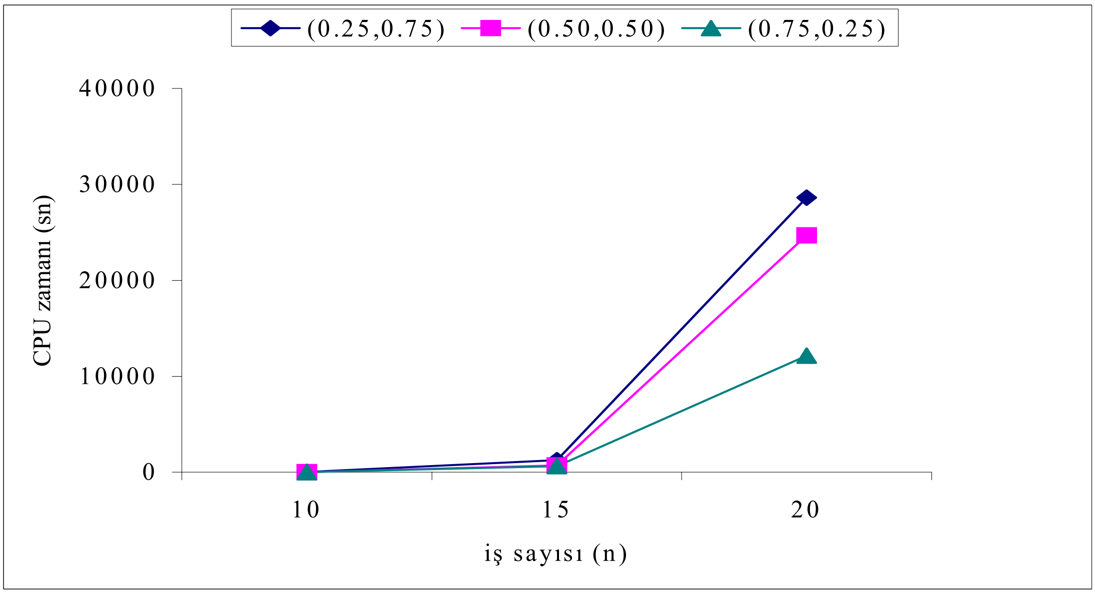
Tablo 7'den görüleceği gibi rassal arama yöntemi diğer yöntemlere göre daha fazla hata oranı vermektedir. Tabu-II ve Tabu-III yöntemleri ise rassal arama ve Tabu-I' e nispeten daha az hata oranları vermektedir ki bu durumda Tabu arama yönteminin başlangıç çizelgesine duyarlı olduğu söylenebilir.

Sezgisel yöntemlerle büyük boyutlu problemler çözülecektir. Büyük boyutlu problemler için deney seti Tablo 8'de verilmiştir. Tablo 8'de de görüldüğü gibi toplam 300 problem çözülmüştür.

Büyük boyutlu problemlerde sezgisel yöntemlerin hataları (2) nolu ifade ile hesaplanmıştır.

$$\text{hata} = \frac{\text{sezgisel çözüm degeri - eniyi sezgisel çözüm degeri}}{\text{eniyi sezgisel çözüm degeri}} \quad (9)$$

Sezgisellerin hataları ağırlıklara göre Şekil 6-8'de gösterilmiştir. Şekil 6'da $(\alpha,\beta)=(0.25,0.75)$ ağırlık değeri için sezgisellerin hataları gösterilmiştir. Maksimum gecikmeye ilişkin ağırlık değerinin fazla olduğu bu tasarımda Tabu-III yöntemi daha iyi sonuç verirken, ikinci olarak Tabu-II yöntemi gelmektedir. Şekil 7'de ise $(\alpha,\beta)=(0.50,0.50)$ ağırlık değeri için sezgisellerin hataları gösterilmektedir. Bu tasarımda toplam tamamlanma zamanı ve maksimum gecikme ölçütlerine ilişkin ağırlık değerleri aynı olup Rassal arama yöntemi en kötü hata performansı verirken Tabu-II ve Tabu-III yöntemleri arasında bariz bir üstünlük sağlanamamıştır. Şekil 8'de $(\alpha,\beta)=(0.75,0.25)$ ağırlık değeri için sezgisellerin hataları gösterilmektedir. Bu tasarımda toplam tamamlanma zamanına ilişkin ağırlık değeri daha fazla olup yine rassal arama en kötü hata performansı verirken Tabu-II yöntemi daha düşük hata değerleri ile sonuçlanmıştır.



Şekil 4. Ağırlık değerlerine göre işlerin toplam CPU zamanları.

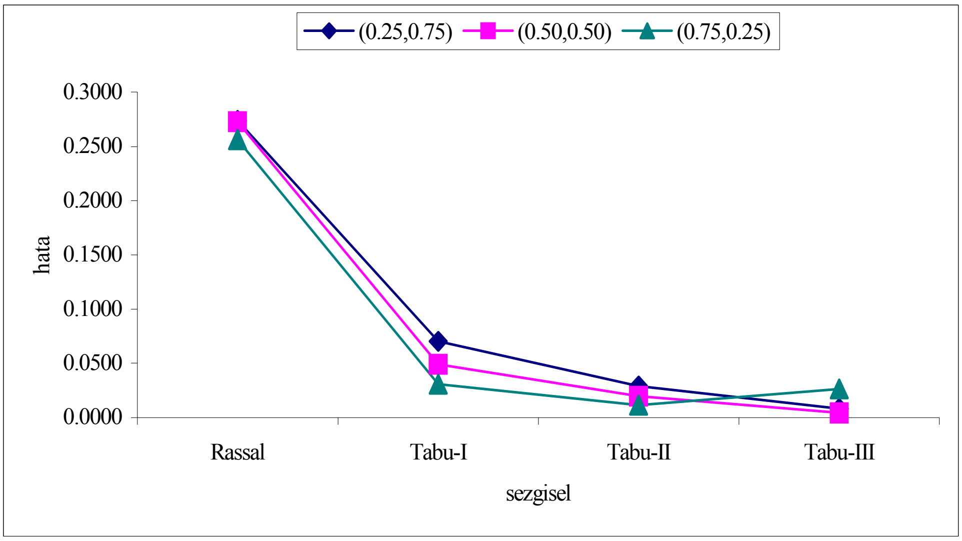
Figure 4. *According to weight values total CPU times of jobs.*

Tablo 6. Tamsayılı programlamanın CPU çözüm süreleri (saniye).
Table 6. *Integer programming solution times(second).*

		(α,β)		
n	(n_1,n_2)	(0.25,0.75)	(0.50,0.50)	(0.75,0.25)
10	(5;5)	4.01	2.84	1.46
	(6;4)	3.84	2.46	1.19
	(7;3)	3.57	2.30	1.10
	(8;2)	3.44	2.29	0.97
	(9;1)	2.91	2.26	0.80
15	(8;7)	288.45	193.01	127.77
	(9;6)	248.38	179.55	118.27
	(10;5)	245.76	173.33	107.00
	(11;4)	241.65	146.06	93.60
	(12;3)	216.40	140.91	90.63
	(13;2)	181.40	116.92	81.61
	(14;1)	154.59	114.85	76.53
20	(10;10)	4166.43	2397.89	1773.30
	(11;9)	3525.23	2084.36	1464.97
	(12;8)	2947.88	1937.63	1197.14
	(13;7)	2880.03	1634.32	1144.78
	(14;6)	2304.58	1497.09	1060.56
	(15;5)	2150.87	1469.98	910.42
	(16;4)	1764.19	1211.94	881.63
	(17;3)	1511.42	1058.71	763.42
	(18;2)	1327.87	995.72	740.08
	(19;1)	1300.74	929.40	649.31

Tablo 7. küçük boyutlu problemlerde sezgisellerin hataları.
Table 7. *Errors of heuristics for small job sizes problems.*

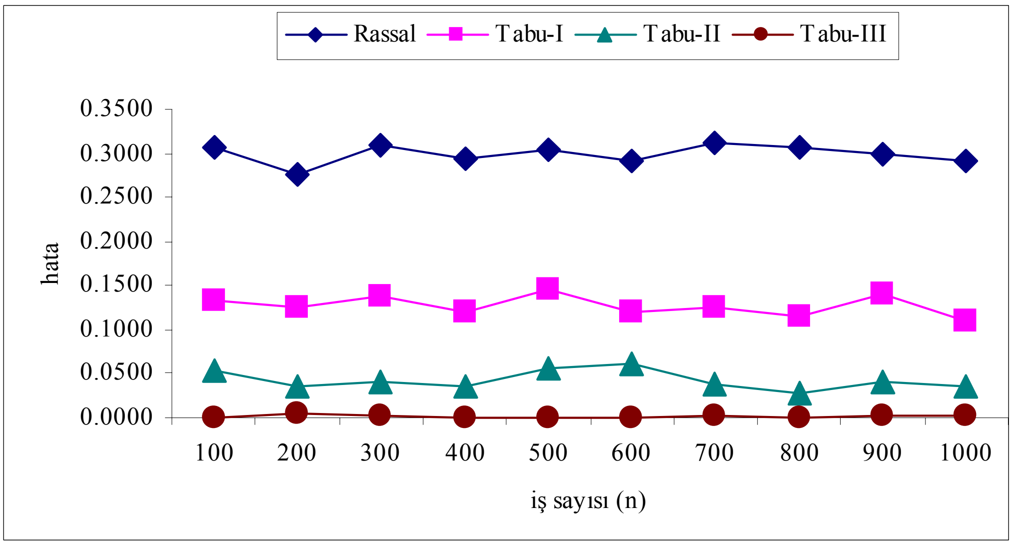
(α,β)	n	Rassal	Tabu-I	Tabu-II	Tabu-III
(0.25,0.75)	10	0.2776	0.0701	0.0447	0.0000
	15	0.2687	0.0693	0.0137	0.0162
	20	0.2753	0.0712	0.0281	0.0078
(0.50,0.50)	10	0.2537	0.0396	0.0104	0.0026
	15	0.2777	0.0498	0.0194	0.0055
	20	0.2864	0.0574	0.0285	0.0038
(0.75,0.25)	10	0.2558	0.0394	0.0113	0.0324
	15	0.2672	0.0281	0.0107	0.0264
	20	0.2461	0.0246	0.0115	0.0196



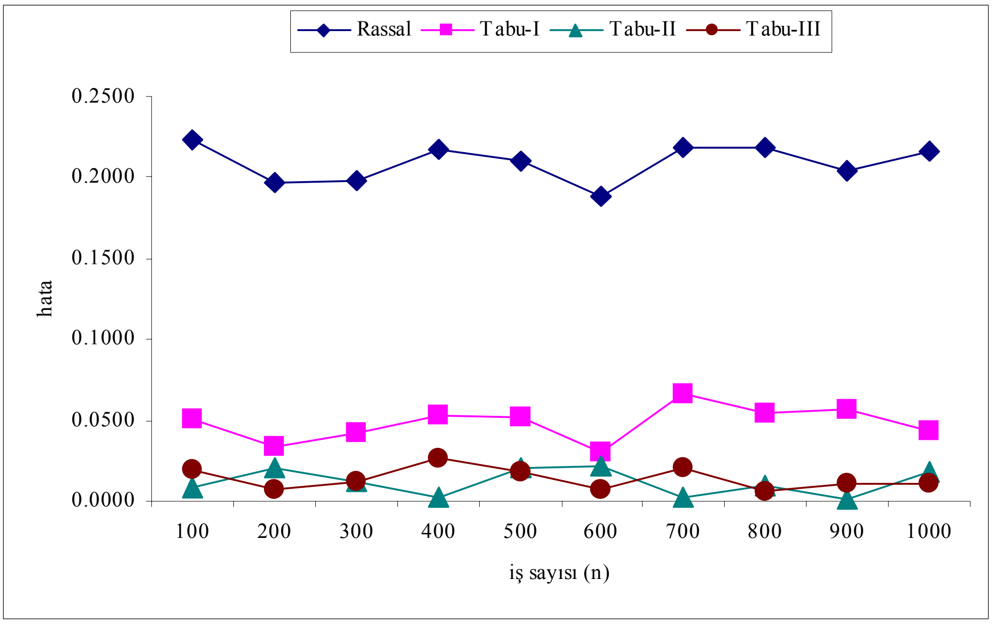
Şekil 5. Sezgisellerin ağırlıklara göre ortalama hata değerleri.
Figure 5. *According to weights average error values of heuristics.*

Tablo 8. Büyük boyutlu problemlerin deney seti.
Table 8. *Experimental set of large size problems.*

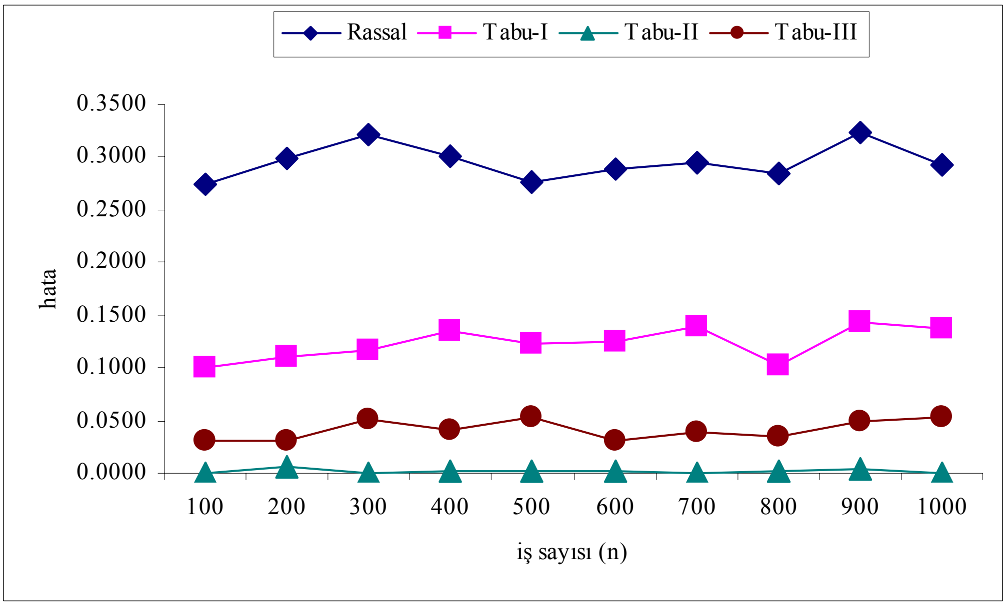
Parametreler	**Alternatif**	**Değerler**
Ağırlıklar (α,β)	3	(0.25,0.75); (0.50,0.50);(0.75,0.25)
İş sayısı, n	10	100,200,…,1000
İşlem zamanı p_j	1	[1,100]
Teslim tarihi d_j	1	$[0, \sum p_j / 2]$
Çözülen problem		10
Toplam problem		3×10×1×1×10=300



Şekil 6. (α,β)=(0.25,0.75) ağırlık değeri için sezgisellerin hataları.
Figure 6. *Heuristics errors for (α,β)=(0.25,0.75) weight value.*



Şekil 7. (α,β)=(0.50,0.50) ağırlık değeri için sezgisellerin hataları.
Figure 7. *Heuristics errors for* (α,β)*=(0.50,0.50)weight value.*

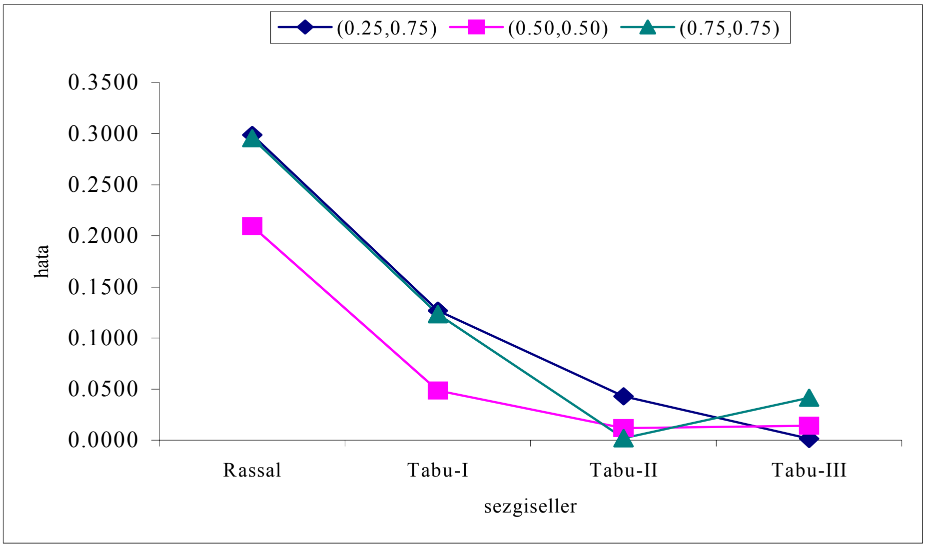


Şekil 8. (α,β)=(0.75,0.25) ağırlık değeri için sezgisellerin hataları.
Figure 8. *Heuristics errors for* (α,β)*=(0.75,0.25) weight value.*

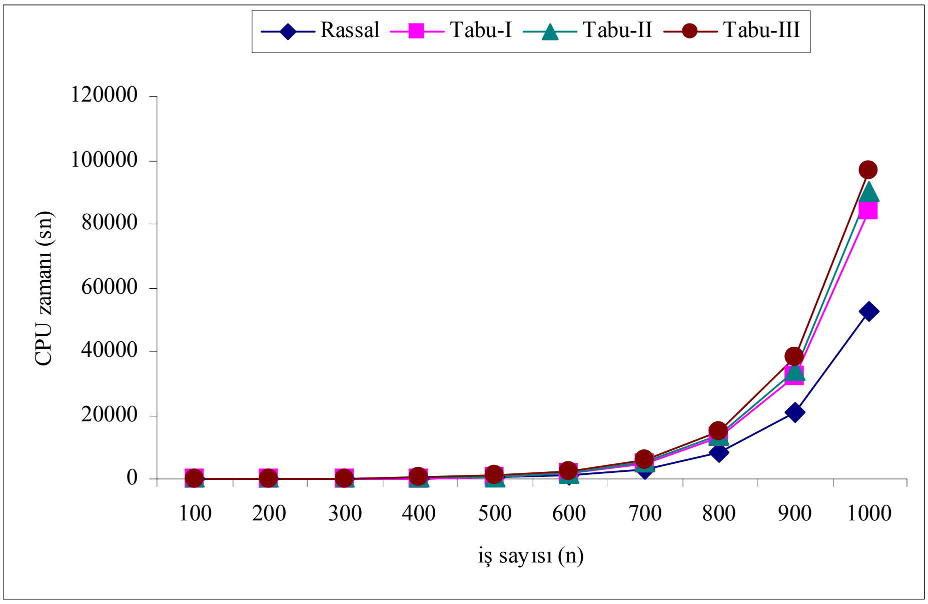
Büyük boyutlu problemlerde sezgisellerin ağırlıklara göre ortalama hata değerleri Şekil 9'de verilmiştir.

Genel olarak toplam tamamlanma zamanı ağırlığı fazla olduğunda SPT kuralına göre elde edilen sırayı, maksimum gecikme değerinin ağırlığı fazla olduğunda ise EDD kuralına göre elde edilen sırayı başlangıç çözüm olarak alındığında tabu arama yönteminin performansını olumlu yönde etkilediği ve daha iyi sonuçlar verdiği görülmüştür.

Büyük boyutlu problemlerin için sezgisellerin çözüm zamanı saniye olarak Şekil 10'da verilmiştir. Tabu arama çözüm sürelerinin birbirine yakın çıktığı rassal aramanın bir miktar daha küçük CPU zamanları verdiği görülmektedir.



Şekil 9. Sezgisellerin büyük boyutlu problemlerde ağırlıklara göre ortalama hata değerleri.
Figure 9. *According to weights average error values of large size problems.*



Şekil 10. Sezgisellerin CPU çözüm zamanı (saniye).
Figure 10. *CPU solution times of heuristics.*

SONUÇ VE ÖNERİLER

Bu çalışmada iki ölçütlü paralel iki özdeş makineli çizelgeleme problemi ele alınmıştır. Ele alınan ölçütler toplam tamamlanma zamanı ve maksimum gecikme olup çizelgeleme literatüründe oldukça önemli olan ölçütlerdir. NP-zor yapıda olan problemin en iyi çözümlerini bulmak için tamsayılı programlama modeli geliştirilmiş, Hyper LINDO/PC 6.01 programı kullanılarak çözümler elde edilmiştir. Geliştirilen bu tamsayılı programlama modeli, $n^3/2+3n^2/2+n/2$ değişkenli ve $3n^2$ kısıtlıdır (n iş sayısını göstermektedir). Tam sayılı programlama yöntemi iş sayısı artıkça üssel olarak artan bir zaman karmaşıklığına sahiptir ve dolayısı ile bu çalışmada 20 işe kadar olan problemlerin en iyi çözümleri üç farklı ağırlık değerinde bulunabilmiştir. Diğer taraftan geliştirilen bu tam sayılı programlama yöntemi problem için önerilen dört sezgisel yaklaşımın (Rassal Arama, Tabu- I, Tabu-II, Tabu- III) performansını belirlemede katkı sağlamıştır. Yani sezgisel yaklaşımlara göre bulunan küçük boyutlu problemlerin sonuçları tamsayılı modelle elde edilen en iyi çözüm sonuçları ile karşılaştırılmıştır. Karşılaştırma sonuçlarına göre küçük boyutlu problemlerde özellikle Tabu-II ve Tabu-III, Rassal Arama ve Tabu I 'e göre çok daha düşük hata oranları vermiştir. Ayrıca büyük boyutlu problemlerin çözümünde de benzer sonuçlar elde edilmiştir. Toplam tamamlanma zamanı ağırlığı fazla olduğunda SPT kuralına göre elde edilen sırayı (Tabu II) , maksimum gecikme değerinin ağırlığı fazla olduğunda ise EDD kuralına göre elde edilen sırayı (TABU III) başlangıç çözüm olarak alındığında tabu arama yönteminin çözüm kalitesini olumlu yönde etkilediği ve daha iyi sonuçlar verdiği görülmüştür.

Bundan sonraki çalışmalarda ikiden çok makineli durumlar incelenebileceği gibi diğer performans ölçütleri de dikkate alınabilir.

KAYNAKLAR

Eck, B.T., Pinedo, M., 1993, On the minimization of the makespan subject to flowtime optimality, *Operations Research*, **41**, 797-800.

Eren, T., 2004, Çok ölçütlü akış tipi çizelgeleme problemleri için çözüm yaklaşımları, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

Eren, T., Güner, E., 2002, Tek ve paralel makineli problemlerde çok ölçütlü çizelgeleme problemleri için bir literatür taraması, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, **17**, 4, 37-69.

Eren, T., Güner, E., 2004, Çok ölçütlü akış tipi çizelgeleme problemleri için bir literatür taraması, *Pamukkale Üniversitesi Mühendislik Fakültesi Mühendislik Bilimleri Dergisi*, **10**, 1, 19-30.

Glover, F., 1986, Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research*, **5**, 533-549.

Gupta, J. N.D., Ho, J.C., Webster, S., 2000, Bicriteria optimization of the makespan and mean flowtime on two identical parallel machines, *Journal of the Operational Research Society*, **51**, 11, 1330-1339.

Gupta, J.N.D., Ho, J.C., 2001, Minimizing makespan subject to minimum flowtime on two identical paralel machines, *Computers and Operations Research*, **28**, 705-717.

Gupta, J.N.D., Ruiz-Torres, A.J., 2000, Minimizing makespan subject to minimum total flow-time on identical parallel machines, *European Journal of Operational Research*, **125**, 370-380.

Lenstra J.K., Kan Rinnooy, A.H.G., Brucker, P., 1977, Complexity of machine scheduling problems, *Annals of Discrete Mathematics*, **4**, 281-300.

Lin, C.H., Liao, C.J., 2004, Makespan minimization subject to flowtime optimality on identical parallel machines, *Computers and Operations Research*, **31**, 10, 1655-1666.

Mohri, S., Masuda, T., Ishii, H., 1999, Bi-criteria scheduling problem on three identical parallel machines, *International Journal of Production Economics*, **60**, 529-536.

Sarin, S.C., Hariharan, R., 2000, Two machine bicriteria scheduling problem, *International Journal of Production Economics*, **65**, 2, 125-139.

Suresh, V., Chaudhuri, D., 1996, Bicriteria scheduling problem for unrelated parallel machines, *Computers and Industrial Engineering*, **30**, 1, 77-82.