İKİ ÖLÇÜTLÜ İKİ MAKİNALI AKIŞ TİPİ ÇİZELGELEME PROBLEMİ

Tamer Eren
Kırıkkale Üniversitesi, Endüstri Mühendisliği Bölümü, 71450, Kırıkkale
Ertan Güner
Gazi Üniversitesi, Endüstri Mühendisliği Bölümü, 06570, Maltepe, Ankara

Özet:Bu çalışmada, çok ölçütlü iki makinalı akış tipi çizelgeleme problemi ele alınmıştır. Ele alınan problemde toplam tamamlanma zamanı ve toplam gecikme ölçütlerinin ağırlıklı toplamının ($n/2/\alpha\sum C+\beta\sum T$) en küçüklenmesi amaçlanmıştır. Literatürde akış tipinde özellikle gecikme ölçütünün yer aldığı çok ölçütlü çalışmaların sayısı oldukça sınırlıdır. NP-zor sınıfında olan bu problemin optimal çözümü için tamsayılı programlama modeli geliştirilmiştir. Verilen modelle iş sayısı 20' ye kadar olan problemlerin çözümü yapılmıştır. Ayrıca büyük boyutlu problemlerin çözümlerini gerçekleştirmek için iki sezgisel yöntem kullanılarak çözüm performansları karşılaştırılmıştır.

Anahtar Kelimeler: *Akış Tipi Çizelgeleme, İki Ölçüt, Tamsayılı Programlama, Sezgisel Yöntemler*

THE BICRITERIA TWO-MACHINE FLOWSHOP SCHEDULING PROBLEM

Abstract: This paper attempts to solve a two-machine flowshop bicriteria scheduling problem, in which the objective function is to minimize a weighted sum of total completion time and total tardiness ($n/2/\alpha\sum C+\beta\sum T$). In literature studies on the multi-criteria flow shop scheduling with tardiness criterion have been very limited. To tackle the problem that is NP hard class, an integer programming model, is proposed The computational results show that the proposed integer programming model is effective in solving problems up to 20 jobs. To solve large size problems, two heuristic methods are also proposed and performances of the methods are compared.

Keywords: *Flowshop Scheduling, Bicriteria, Integer Programming, Heuristic Methods*

1. Giriş

Akış tipi çizelgelemede çalışmaların çoğunluğu tamamlanma zamanı üzerinde odaklanan problemlerdir. Ancak çizelgelemede önemli etkiye sahip ve müşteri memnuniyetinin bir göstergesi olan toplam gecikme ölçütü üzerinde yapılmış çalışma sayısı oldukça sınırlıdır. Bu iki ölçütün birlikte dikkate alındığı tek çalışma ise Lee ve Wu (2001) tarafından yapılmış ve dal-sınır yöntemi kullanılarak problem çözülmüştür. Bu çalışmada toplam tamamlanma zamanı ve toplam gecikmenin ağırlıklı toplamını enküçüklendiği iki makinalı akış tipi bir çizelgeleme problemi dikkate alınmıştır. Problemlerin çözümü için bir tamsayılı programlama modeli geliştirilmiştir.

2. Tamsayılı Programlama Modeli

Önerilen tamsayılı programlama modeli n^2+4n değişkenli 5n kısıtlıdır (n: iş sayısı). Modelde kullanılan varsayımlar; hazırlık zamanları biliniyor ve işlem zamanına dahil edilmiştir, başlanan bir iş makinada bitiriliyor, makinaların çizelgeleme periyodunca bozulmayacağı ve bir makinada aynı anda birden fazla iş yapılamayacağı kabul edilmiştir. α ve β toplam tamamlanma zamanı ve toplam gecikme ağırlığını p_{ji}; i. makinada makinada j işinin işlem zamanı, d_j ise j işinin teslim tarihini göstermektedir.

Karar Değişkenleri

Z_{jk} : eğer *j* işi *k.* sırada işlem görmek için çizelgelenmişse 1, aksi halde 0, *j,k*=1, 2, ..., *n*

X_k : ikinci makinadaki *k.* sıradaki işin başlangıcı ve *(k-1).* sıradaki işin bitimi arasındaki boş zaman, *k*=1, 2, ..., *n*

Y_k : *k.* sıradaki iş için o işin birinci makinada bitişi ve ikinci makinada işlemin başlaması arasındaki zaman dilimi, *k*=1, 2, ..., *n*

S_k : *k.* sıradaki işin birinci makinada başlangıç zamanı, *k*=1, 2, ..., *n*.

T_k : *k.* sıradaki işin gecikmesini göstermektedir.

$T_k = \max\{C_k-d_k,0\}$ veya $T_k \geq C_k-d_k$ ve $T_k \geq 0$ *k*=1, 2, ..., *n*.

Yardımcı Değişkenler

A_k : Birinci makinada *k.* sıradaki işin işlem zamanı $A_k=\sum Z_{jk}p_{j1}$ $k=1, 2, ..., n.$

B_k : *ikinci* makinada *k.* sıradaki işin işlem zamanı $B_k=\sum Z_{jk}p_{j2}$ $k=1, 2, ..., n.$

C_k : *ikinci* makinada *k.* sıradaki işin tamamlanma zamanı $C_k=\sum X_j+\sum B_j$ $k=1, 2, ..., n.$

veya $C_1=X_1+B_1$ ve $C_k=C_{k-1}+X_k+B_k$ $k=2, 3, ..., n.$

C_{max} : *son işin* tamamlanma zamanı $C_{max}=C_n=\sum_j X_j+\sum_j B_j$ $k=1, 2, ..., n.$

d_k* : *k.* sıradaki işin teslim tarihi $d_k^*=\sum Z_{jk}d_j$ $k=1, 2, ..., n.$

Önerilen Model

Amaç fonksiyonu:

$$Min\ Z=\alpha\sum_{k=1}^{n}C_k+\beta\sum_{k=1}^{n}T_k$$

Kısıtlar:

$$\sum_{j=1}^{n}Z_{jk}=1 \qquad k=1, 2, ..., n. \qquad (1)$$

$$\sum_{k=1}^{n}Z_{jk}=1 \qquad j=1, 2, ..., n. \qquad (2)$$

$$S_k\geq S_{k-1}+A_{k-1} \qquad k=2, 3, ..., n. \qquad (3)$$

$$X_1=S_1+A_1+Y_1;\qquad X_k=S_k+A_k+Y_k-C_{k-1} \qquad k=2, 3, ..., n. \qquad (4)$$

$$T_k\geq C_k-d_k^* \qquad k=1, 2, ..., n. \qquad (5)$$

$Z_{jk}=0-1$ ve diğer tüm değişkenler 0'dan büyük ve eşit

Kısıt (1); *k.* iş önceliğinde sadece bir tek iş çizelgelenmesini, kısıt (2); her bir işin sadece bir kez çizelgelenmesini ifade etmektedir. Kısıt (3); birinci makinedeki *k.* sıradaki işin işleme başlama zamanının önceki işlerin bitiş zamanından büyük veya eşit olması gerektiğini göstermektedir. Kısıt (4); ikinci makinedeki *k.* sıradaki iş ile *(k-1).* sıradaki iş arasında makinanın boş bekleme zamanını ifade etmektedir. Kısıt (5); k. Sırada ki işin gecikmesini ifade etmektedir.

3. Tabu Arama Yöntemi

Tabu arama yöntemi kombinatoryal problemlerde kullanılan sezgisel optimizasyon tekniklerinden 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.

4. Deneysel Sonuçlar

Yapılan çalışmada bütün deneysel testler Pentium IV/2 GHz 512 RAM kapasiteli kişisel bilgisayar kullanılmıştır. İşlem zamanları 1 ile 10 arasında düzgün dağılımdan üretilmiştir. Teslim tarihleri P(1-τ-R/2) ve P(1-τ+R/2) aralığında alınmıştır. P= $\sum_j p_{j2}+\min\{p_{j1}\}$ olmak üzere, τ=0.25, 0.50, 0.75 ve R= 0.25, 0.50, 0.75, 1.00 olarak 12 durumda incelenmiştir. Her problem τ ve R'nin kombinasyonlarında 10 değişik problemle çözülmüştür. Her problemde n = 10, 12, 14, 16, 18 ve 20 olmak üzere altı durum olmak üzere 720 problem çözülmüştür. α =0.5 ve β=0.5 değerleri için iş sayılarına göre CPU çözüm zamanları Tablo 1'de verilmiştir. Tablo 1'de görüldüğü gibi iş sayısı arttıkça işlem zamanı büyümektedir. (τ,R)=(0.50,1.00) ve (0.75,1.00) değerlerinde problem çözümü zorlaşmaktadır.

Tablo 1. Optimal yöntemin çözüm sonuçları

n	τ	R	CPU(sn)	n	τ	R	CPU(sn)	n	τ	R	CPU(sn)	n	τ	R	CPU (sn)
10	0.25	0.25	0,70	12	0.50	0.75	19,30	16	0.25	0.25	26,40	18	0,50	0.75	491,10
		0.50	0,10			1.00	26,60			0.50	24,20			1.00	890,80
		0.75	0,50		0.75	0.25	12,10			0.75	405,80		0.75	0.25	344,20
		1.00	3,60			0.50	13,40			1.00	163,50			0.50	625,60
	0.50	0.25	0,90			0.75	33,10		0.50	0.25	48,40			0.75	762,00
		0.50	0,90			1.00	13,30			0.50	75,70			1.00	692,70
		0.75	10,50	14	0.25	0.25	7,30			0.75	184,90	20	0.25	0.25	498,70
		1.00	8,20			0.50	14,40			1.00	271,40			0.50	527,40
	0.75	0.25	1,40			0.75	6,40		0.75	0.25	71,40			0.75	547,00
		0.50	1,10			1.00	87,20			0.50	205,70			1.00	1283,00
		0.75	1,60		0.50	0.25	4,00			0.75	292,80		0.50	0.25	587,10
		1.00	2,50			0.50	14,80			1.00	222,50			0.50	831,50
12	0.25	0.25	0,90			0.75	81,30	18	0.25	0.25	109,70			0.75	1247,60
		0.50	1,00			1.00	125,30			0.50	98,60			1.00	1820,20
		0.75	1,40		0.75	0.25	31,50			0.75	221,70		0.75	0.25	983,90
		1.00	15,50			0.50	53,50			1.00	371,00			0.50	1448,80
	0.50	0.25	1,70			0.75	93,10		0.50	0.25	182,60			0.75	1771,90
		0.50	10,10			1.00	68,70			0.50	239,70			1.00	1792,20

Problem etkin bir şekilde optimal olarak ancak 20 işe kadar çözülebilmektedir. Daha büyük boyutlu problemleri çözmek için NEH (Nawaz vd., 1983) yöntemi probleme uyarlanmıştır. NEH yöntemi ayrıca geliştirilen tabu arama yönteminde başlangıç çözümü olarakta kullanılmıştır. Tabu arama için yapılan deneysel tasarımda komşu arama stratejisi olarak bitişik iş çiftlerinin yerdeğiştirmesi (API) metodu, tabu listesi uzunluğu $2\sqrt{n}$ ve durdurma koşulu olarak da *n* tekrarda iyileşmeme koşulu dikkate alınmıştır. Sezgisel yöntemlerin çözüm kalitesini belirlemek için Çözüm kalitesi (ÇK) = (2×optimal çözüm-sezgisel çözüm) / optimal çözüm formulü kullanılmıştır. Tablo 2'de NEH yöntemiyle tabu arama yönteminin çözüm kalitesi açısından karşılaştırılması verilmektedir. Tabloya göre NEH'in çözüm kalitesi %99'a yakın bulunurken, tabu aramanın çözüm kalitesi %99,5 bulunmuştur. Ayrıca Tablo 3'de büyük boyutlu problemler 100'den 1000'e kadar 10 değişik durumda 10 problem olmak üzere toplam 100 problem çözülmüştür.bu problemlerde tabu arama çözümü en iyi çözüm kabul edilerek NEH yönteminin çözüm kalitesi ile karşılaştırılmıştır. İş sayısı arttıkça NEH'in sonuçları tabu aramaya yaklaştığı görülmektedir. Ayrıca NEH ile 100 işli problem yaklaşık 102 saniyede çözülürken tabu aramayla 1071 saniyede çözülmektedir.

Tablo 2 .Sezgisellerin çözüm kalitesi

N	NEH Ç.K	NEH süre (sn)	Tabu Ç.K.	Tabu süre (sn)
10	0,9905	<0,01	0,9963	0,0159
12	0,9955	<0,01	0,9971	0,0199
14	0,9837	<0,01	0,9963	0,0299
16	0,9808	<0,01	0,9914	0,0406
18	0,9896	<0,01	0,9945	0,0562
20	0,9862	<0,01	0,9928	0,0642
ort	0,9877	<0,01	0,9947	0,0378

Tablo 3 .Büyük boyutlu problemlerin çözümü

n	NEH Ç.K.	NEH CPU (sn)	Tabu CPU (sn)
100	0,9938	0,485	2,922
200	0,9969	2,141	12,485
300	0,9924	5,312	38,718
400	0,9949	10,422	60,875
500	0,9986	17,718	90,828
600	0,9977	27,141	154,484
700	0,9992	41,547	311,328
800	0,9993	57,235	450,406
900	0,9993	78,422	663,531
1000	0,9995	101,797	1070,843

5. Sonuç

Bu çalışmada toplam tamamlanma zamanı ve toplam gecikmenin ağırlıklı toplamını enküçüklemek için tamsayılı programlam yaklaşımı geliştirilmiştir. Tamsayılı programlama modeli ile 20 işe kadar olan problemlerin optimal çözümleri elde edilmiştir. Uyarlanan NEH yöntemi ve tabu arama yönteminin optimum değere çok yakın sonuçlar verdiği gösterilmiştir. Ayrıca bu yöntemlerle 1000 işe kadar olan problemlerin çözümü gerçekleştirilmiştir.

Kaynaklar

Lee, W. – C., and Wu, C. – C., Minimizing the total flow time and the tardiness in a two-machine flowshop, *International Journal of Systems Science* 32 (3), 365-373, 2001

Nawaz, M., vd., A heuristic algorithm for the m-machine, n-job flowshop sequencing problem, *OMEGA,* 11, 91–95 1983.