

SIRA BAĞIMLI HAZIRLIK SÜRELİ İKİ ÖLÇÜTLÜ TEK MAKİNE ÇİZELGELEME PROBLEMİ İÇİN SEZGİSEL BİR ÇÖZÜM YÖNTEMİ

Feriştah ÖZÇELİK*, Tuğba SARAÇ

Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü,
Meşelik Yerleşkesi, 26480 Eskişehir
fdurmaz@ogu.edu.tr, tsarac@ogu.edu.tr

Geliş Tarihi: 4 Kasım 2009; Kabul Ediliş Tarihi: 23 Kasım 2011
Bu makale 3 kez düzeltilmek üzere 146 gün yazarlarda kalmıştır.

ÖZET

Bu çalışmada sıra bağımlı hazırlık süreli, son işin tamamlanma zamanını ve toplam gecikmeyi en aza indirmeyi amaçlayan tek makine çizelgeleme problemi ele alınmıştır. Bu problem NP-zor sınıfında yer almaktadır. Problemin çözümü için ilk olarak, SST (en kısa hazırlık süresi) sıralama kuralının ağırlıklandırılmış hali olan yeni bir sıralama kuralı (WSST) önerilmiştir. Bu sıralama kuralı, literatürde yer alan EDD (en küçük teslim zamanı), SST ve ATCS (Apparent Tardiness Cost with Setups) sıralama kurallarıyla karşılaştırılmıştır. Daha sonra, önerilen sıralama kuralının çözümlerini, bir yerel arama yöntemiyle iyileştirme amacını güden yeni bir sezgisel algoritma önerilmiştir. Önerilen yöntemlerin etkinliği rassal olarak türetilen test problemleri kullanılarak araştırılmıştır.

Anahtar Kelimeler: Tek makine çizelgeleme problemi, sıra bağımlı hazırlık süresi, sezgisel yöntemler

A HEURISTIC FOR BICRITERIA SINGLE MACHINE SCHEDULING PROBLEM WITH SEQUENCE DEPENDENT SETUP TIMES

ABSTRACT

In this study, the bi-criteria scheduling problem of minimizing the makespan and total tardiness on a single machine with sequence dependent setup times is considered. This problem is known as NP-hard. To solve this problem, firstly we propose a new dispatching rule named as WSST that is the weighted form of the SST (Shortest Setup Time) rule. The proposed rule is compared with EDD (Earliest Due Date), SST and ATCS (Apparent Tardiness Cost with Setups) dispatching rules. And then, a new heuristic which improves the solutions of the WSST by a local search algorithm is proposed. The performance of the proposed heuristics is tested by using randomly generated test problems.

Keywords: Single machine scheduling problem, sequence dependent setup time, heuristics

* İletişim yazarı

1. GİRİŞ

Üretimi gerçekleştirmek için makine veya süreç üzerinde yapılan işlemlere hazırlık faaliyetleri denilir. Bunlar; gerekli ekipmanların temini, ayarlanması, takılması, temizlenmesi gibi faaliyetlerden oluşur. Hazırlık işlemleriyle ilgili problemler literatürde iki sınıfta ele alınmıştır. Birincisinde, hazırlıklar sadece işlem görecektir işe bağlı olup sıra bağımsız hazırlık süresi olarak ifade edilir. Diğerinde ise hazırlık, hem o anda işlem görecektir işe hem de bir önceki işe bağlıdır. Bu durum ise sıra bağımlı hazırlık süresi olarak ifade edilir. Sıra-bağımlı hazırlık süresi uygulamalarıyla ilgili örnek olarak matbaa endüstrisi verilebilir. Burada makinenin temizlenmesi ve hazırlanması, en son kullanılan mürekkep rengine, kağıdın boyutuna ve özelliğine bağlıdır. Sıra bağımlı hazırlık sürelerine kimya, ilaç, metal ve plastik endüstrilerinde de sıkça rastlanmaktadır.

Literatürde sıralamaya bağlı hazırlık sürelerinin göz önünde bulundurulduğu çok sayıda çalışma mevcuttur (Tan ve Narasimhan, 1997; Wang ve Wang, 1997; Kolahan ve Liang, 1998; Asano ve Ohta, 1999; Miller vd., 1999; Armentano ve Mazzini, 2000; Tan vd., 2000; Franca vd., 2001; Gagne vd., 2002; Mendes vd., 2002; Shin vd., 2002; Chang vd., 2004; Lee ve Aslani, 2004; Rabadi vd., 2004; Eren ve Guner, 2006; Gupta ve Smith, 2006). Ancak bu çalışmaların çok azında problem çok amaçlı olarak ele alınmıştır (Kolahan ve Liang, 1998; Lee ve Aslani, 2004; Rabadi vd., 2004; Eren ve Guner, 2006; Choobineh vd., 2006).

Kolahan ve Liang (1998) sıra bağımlı hazırlık sürelerinin söz konusu olduğu ve işlem sürelerinde azaltma ve arttırmanın doğrusal bir maliyeti olduğu tam zamanında çizelgeleme problemi için bir yasaklı arama algoritması önermişlerdir. Amaç fonksiyonu, toplam ağırlıklı erken bitirme, gecikme ve toplam ağırlıklı işlem sürelerinde azaltma ve arttırma maliyetlerinin doğrusal bileşimidir.

Lee ve Aslani (2004), birincil amacın geciken iş sayısının en aza indirilmesi, ikincil amacın da en son işin tamamlanma zamanının en aza indirilmesi olduğu problem için 0-1 karma tamsayı doğrusal bir matematiksel model önermişlerdir.

Rabadi vd. (2004), toplam erken bitirme ve toplam gecikmenin doğrusal bileşiminden oluşan bir amaç fonksiyonuna sahip problemin çözümüne yönelik olarak bir dal sınır algoritması önermişlerdir. Önerilen yöntemle 25 işe kadar olan problemleri makul süreler içinde çözebildiklerini göstermişlerdir.

Eren ve Güner (2006) toplam tamamlanma zamanı ve toplam gecikmeyi en aza indirme amacını ele almışlardır. Problem için bir tamsayı matematiksel model geliştirmişlerdir. Problemin çözümü için basit bir sezgisel yöntem önermiş ve bu yöntemle elde ettikleri çözümü, önerdikleri ikinci çözüm yaklaşımı olan yasaklı arama algoritmasının başlangıç çözümü olarak kullanmışlardır.

Choobineh vd. (2006) sıra bağımlı hazırlık süreli tek makine çizelgeleme probleminde; son işin tamamlanma zamanı, geciken iş sayısı ve toplam gecikme amaçlarını dikkate almışlardır. Bir karma tamsayı matematiksel model geliştirmişler ve problemin çözümü için çok amaçlı yasaklı arama yöntemini önermişlerdir.

Bu çalışmada sıra bağımlı hazırlık süreli, son işin tamamlanma zamanını (C_{max}) ve toplam gecikmeyi (ΣT) en aza indirmeyi amaçlayan tek makine çizelgeleme problemi ele alınmıştır. NP-zor sınıfında yer alan bu çok amaçlı problemin çözümü için, en kısa hazırlık süresi (SST) sıralama kuralının ağırlıklandırılmış hali olan yeni bir sıralama kuralı (WSST) önerilmiştir. Bu sıralama kuralı, literatürde yer alan en küçük teslim zamanı (EDD), SST ve ATCS (*Apparent Tardiness Cost with Setups*) sıralama kurallarıyla karşılaştırılarak başarısı ortaya konmuştur. Daha sonra önerilen sıralama kuralının çözümlerini bir yerel arama yöntemiyle iyileştirme mantığını taşıyan yeni bir sezgisel algoritma önerilmiştir.

Çalışmanın izleyen bölümünde ele alınan problem tanımlanmış, üçüncü bölümde amaçları birleştirme yöntemi açıklandıktan sonra, dördüncü bölümde önerilen çözüm yöntemleri tanımlanmıştır. Beşinci bölümde deneysel sonuçlar tartışılmış ve son bölümde ise, elde edilen sonuçlar yorumlanmıştır.

2. PROBLEMİN TANIMLANMASI VE MODELLENMESİ

Ele alınan problemde tek makinada işlem görecektir n tane iş ($j=1, 2, \dots, n$) sıfırıncı zamanda işlem için hazırır. p_j , j işinin işlem süresini d_j , j işinin teslim zamanını göstermektedir. Başlangıç hazırlık süresi (h_j) ilk işin tamamlanma zamanına eklenecektir. Sıra bağımlı hazırlık süreleri de k . ($k > 1$) sıradaki işin tamamlanma zamanına eklenecektir. İşler tek makine üzerinde kesintisiz olarak işlem görmekte olup, makine üzerinde birim zamanda sadece tek bir işin işlemi yapılabilmektedir. Bir iş, teslim zamanından sonra biterse o iş gecikmiş olur. Bu çalışmada ele alınan amaçlar son işin tamamlanma zamanının (C_{max}) ve toplam gecikmenin ($\sum T$) en aza indirilmesidir. Bu problem literatürde $(1/s_{ij}/C_{max}, \sum T)$ olarak gösterilmektedir. Problemin matematiksel modeli aşağıda verilmiştir:

Kümeler

$N = \{1, 2, \dots, n\}$ iş kümesi

İndisler

i veya $j \in N$ iş indisi

$k \in N$ sıra indisi

Kısıtlar

$$C_j + M(1 - y_{jk}) \geq h_j + p_j \quad \forall j, k = 1, j, k \in N \quad (3)$$

$$C_j - C_i + M(1 - z_{ijk}) \geq s_{ij} + p_j \quad \forall i \neq j, k > 1, i, j, k \in N \quad (4)$$

$$1 + z_{ijk} \geq y_{ik-1} + y_{jk} \quad \forall i \neq j, k > 1, i, j, k \in N \quad (5)$$

$$\sum_{j \in N} y_{jk} = 1 \quad \forall k \in N \quad (6)$$

$$\sum_{k \in N} y_{jk} = 1 \quad \forall j \in N \quad (7)$$

$$T_j \geq C_j - d_j \quad \forall j \in N \quad (8)$$

$$C_{max} \geq C_j \quad \forall i \neq j, k = n \in N \quad (9)$$

$$y_{jk} \in \{0,1\} \quad \forall j, k \in N \quad (10)$$

$$z_{ijk} \in \{0,1\} \quad \forall i \neq j, k \in N \quad (11)$$

$$C_j \geq 0 \quad \forall j \in N \quad (12)$$

$$C_{max} \geq 0 \quad (13)$$

$$T_j \geq 0 \quad \forall j \in N \quad (14)$$

Parametreler

n : iş sayısı

p_j : j işinin işlem süresi

d_j : j işinin teslim zamanı

h_j : ilk sıradaki işin (j) sıra bağımlı hazırlık süresi (başlangıç hazırlık süresi)

s_{ij} : i işi j işinden önce sıralandığında j işinin sıra bağımlı hazırlık süresi

M : çok büyük bir pozitif tamsayı

Karar değişkenleri

y_{jk} : j işi k . sıraya atanırsa 1; diğer durumda 0

z_{ijk} : j işi k . sıraya atanırsa ve i işinden hemen sonra yapılacaksa 1; diğer durumda 0

C_j : j işinin tamamlanma zamanı

C_{max} : son işin (sıralamada n . sıradaki işin) tamamlanma zamanı

T_j : j işinin gecikmesi $T_j = \text{enb}\{C_j - d_j, 0\}$

Amaç fonksiyonları

$$\text{Enk F1} = C_{max} \quad (1)$$

$$\text{Enk F2} = \sum_{j=1}^n T_j \quad (2)$$

Tablo 1. Tek Amaçlı Problemlerin GAMS Çözümleri

problem	amaç	süre (sn)	F1	F2
10_2_1	F1	10000	1208	516
10_2_1	F2	5953	1255	231
20_2_1	F1	10000	2553	4224
20_2_1	F2	-	-	-
30_2_1	F1	10000	4172	10868
30_2_1	F2	10000	4184	10880
150_2	F1	-	-	-
150_2	F2	-	-	-

(1) ve (2) amaç fonksiyonlarını göstermektedir. Amaç (1) son işin tamamlanma zamanını ve amaç (2) toplam gecikmeyi en aza indirmektedir. Kısıt (3) ilk işin ve kısıt (4) k . ($k > 1$) sıradaki işin tamamlanma zamanını belirlemektedir. Kısıt (5) z_{ijk} karar değişkeninin 1 veya 0 değerini almasını sağlamaktadır. Kısıt (6) ve (7) sırasıyla her konuma yalnızca bir işin atanmasını ve bir işin yalnızca bir konuma atanmasını sağlamaktadır. Kısıt (8) j işinin gecikmesini ve kısıt (9) son işin tamamlanma zamanını belirlemektedir. Kısıt (10) - (14) karar değişkenlerine ait işaret kısıtlarıdır.

Beşinci bölümde özellikleri açıklanmış olan 10, 20, 30 ve 150 işli dört problem, GAMS paket programının CPLEX çözücüsü kullanılarak çözülmüş ve sonuçlar Tablo 1'de sunulmuştur. Çözüm süresi 10.000 sn. ile sınırlandırılmış ve her bir problem C_{max} (F1) ve $\sum T$ (F2) amaçları için ayrı ayrı tek amaçlı olarak çözülmüştür. Tablonun ilk sütununda problemin adı, izleyen sütunlarda sırasıyla dikkate alınan amaç, çözüm süresi ve her iki amaç fonksiyonunun aldığı değerler (F1, F2) verilmiştir.

Tablodan da görülebileceği gibi matematiksel modelin GAMS/CPLEX çözücüsüyle çözümü araştırıldığında, sadece büyük boyutlu problemlere değil bazı küçük boyutlu problemlere bile izin verilen sürede çözüm bulabilmek mümkün olamamıştır. Bu nedenle bu çalışmada, büyük boyutlu problemleri makul sürelerde çözebilecek yeni çözüm yaklaşımları üzerinde durulmuştur.

3. AMAÇLARIN BİRLEŞTİRİLMESİ

Amaçların birleştirilmesi (*scalarization*), birden fazla amaç fonksiyonunun hepsini temsil edebilecek tek bir fonksiyona dönüştürülmesidir. Literatürde birçok amaç birleştirme yöntemi mevcuttur (Luc, 1989; Chankong ve Haimes, 1983; Ehrgott, 2005). Dışbükey olmayan problemler için ϵ -kısıt, konik skalerleştirme, Tchebycheff gibi amaç birleştirme yöntemleri mevcuttur. Bu çalışmada amaçların birleştirilmesi için yaygın ve başarılı kullanımı dikkate alınarak genişletilmiş (augmented) ağırlıklı Tchebycheff fonksiyonu kullanılmıştır. Bu fonksiyon aşağıda verilmiştir.

$$\text{Enk}_{x \in X} \text{Enb}_{l=1..m} [w_l(f_l - y_l^U)] + \rho \sum_{l=1}^m (f_l - y_l^U)$$

Burada ρ , yeterince küçük pozitif bir sayıdır. f_l , l . amaç fonksiyonu y_l^U , l . amaç fonksiyonu için referans değeri ve m amaç sayısıdır.

4. ÖNERİLEN ÇÖZÜM YÖNTEMLERİ

İncelenen çok amaçlı çizelgeleme probleminin çözümü için, SST sıralama kuralının ağırlıklandırılmış hali olan yeni bir sıralama kuralı (WSST) önerilmiştir. Daha sonra önerilen sıralama kuralının çözümlerini bir yerel arama yöntemiyle iyileştirme mantığını taşıyan yeni bir sezgisel algoritma önerilmiştir. Bu algoritmalar izleyen bölümlerde açıklanmaktadır.

4.1 WSST Sıralama Kuralı

Tek makina çizelgeleme problemlerinde kullanılacak pek çok sıralama kuralı vardır. Problemin amacına bağlı olarak bu kuralların başarıları da farklılık

göstermektedir. Son işin tamamlanma zamanı (C_{max}) amacı için önerilen sıralama kuralı en kısa hazırlık süresi (SST); toplam gecikmeyi (ΣT) en aza indirme amacı için ise en küçük teslim zamanı (EDD) sıralama kuralı kullanılmıştır. (Pinedo, 2002). EDD sıralama kuralında işler, teslim zamanlarına göre küçükten büyüğe sıralanır. SST kuralında ise bir iş tamamlandığında bu işten, en kısa hazırlık süresi gerektiren işe geçilir. Sıra bağımlı hazırlık süreleri söz konusu olduğunda kullanılabilir kurallardan bir diğeri de ağırlıklı gecikmelerin toplamı amacı için önerilen ATCS (*Apparent Tardiness Cost with Setups*) sıralama kuralıdır. Sözü geçen sıralama kuralları, genellikle tek bir amaç için başarılı sonuçlar üretirken, birden fazla amaç aynı anda ele alındığında aynı başarıyı yakalamaları zordur. Bu nedenle, bu çalışmada C_{max} ve ΣT amaçlarını birlikte dikkate alan ağırlıklandırılmış en kısa hazırlık süresi (WSST) sıralama kuralı önerilmiştir. Önerilen yaklaşımda, en küçük başlangıç hazırlık süresine ($enk_j\{h_j\}$) sahip olan iş ilk sıraya yerleştirilir. i . iş yerleştirilmişken sonrasında hangi işin (j) geleceğine karar vermek için yerleştirilebilecek işler arasından en küçük ağırlıklandırılmış hazırlık süresine ($enk_j\{d_j s_{ij}\}$) sahip olan iş seçilir. Burada hazırlık sürelerinin teslim zamanları kullanılarak ağırlıklandırılması her iki amacın birlikte ele alınmasını sağlamıştır.

4.2 İyileştirilmiş WSST

İyileştirilmiş WSST (i-WSST) algoritmasında WSST ile elde edilen çözüm 2-opt arama algoritması ile iyileştirilmektedir.

2-opt arama algoritması üç adımdan oluşmaktadır:

Adım 1. S uygun bir başlangıç çözüm ve z onun amaç fonksiyonu değeri olsun (Bu çalışmada başlangıç çözüm, WSST ile bulunmuştur). $S^* = s$, $z^* = z$, $i = 1$ ve $j = i + 1 = 2$ olsun.

Adım 2. i . ve j . sıradaki işlerin yerini karşılıklı olarak değiştir. Bu değişiklik sonucunda ortaya çıkan çözüm S' ve onun amaç fonksiyonu değeri z' iken $z' < z^*$ ise $z^* = z'$ ve $S^* = S'$ olarak al. n , iş sayısı iken eğer $j < n$ ise Adım 2'yi tekrarla; değilse $i = i + 1$ ve $j = i + 1$ olarak al. Eğer $i < n$ ise Adım 2'yi tekrarla; değilse Adım 3'e git.

Adım 3. Eğer $S \neq S^*$ ise $S = S^*$, $z = z^*$, $i = 1$, $j = i + 1 = 2$ olarak al ve Adım 2'ye git. Değilse, S^* eniyi çözümdür, DUR.

2-opt arama algoritması yalnızca ikili değişiklikleri dikkate almaktadır. Algoritma ilk olarak birinci ve ikinci sıradaki işlerin yerini karşılıklı olarak değiştirmektedir. Bu değişiklik amaç fonksiyonunun değerinde bir iyileşme sağladıysa bu çözüm saklanmaktadır. Eğer iyileşme sağlamadıysa bu çözüm göz ardı edilir ve algoritma birinci sıradaki iş ile üçüncü sıradaki işin yerini değiştirir. Bu değişiklik daha iyi bir çözüm türetti ise bu çözüm saklanır, aksi hâlde göz ardı edilir. O hâlde, daha iyi bir çözüm buldukça bir önceki eniyi çözüm göz ardı edilir. Bu süreç, bütün ikili değişiklikler deneninceye kadar devam eder. Toplamda n iş olduğu ve her bir iş, $(n - 1)$ işle yer değiştirebileceği için $n(n - 1) / 2$ tane ikili değişiklik mümkündür. Bu $n(n - 1) / 2$ değişiklik Adım 2'de dikkate alınmaktadır. Adım 2'nin sonunda elde edilen en iyi çözüm, amaç fonksiyonunun değerinde en çok iyileşme sağlayan seçenektir. Bu çözüm başlangıç çözüm olarak alınarak Adım 2 tekrarlanır. Algoritma en iyi çözümde bir değişiklik olmayıncaya kadar devam eder.

5. DENEYSEL SONUÇLAR

Bu bölümde önerilen çözüm yöntemlerinin başarıları tartışılmaktadır. Test problemlerinin nasıl üretildiği açıklandıktan sonra, amaç fonksiyonları için referans noktalarını belirleme yöntemi ve elde edilen test sonuçları sunulmuştur.

5.1 Test Problemlerinin Türetilmesi

Test problemleri, Rajendran ve Ziegler (2003)'in yöntemine benzer bir şekilde türetilmiştir. İşlem süreleri (p_j) [1, 99] aralığında düzgün dağılıma uygun türetilmiştir. Sıra bağımlı hazırlık süreleri (s_{ij}) [1, 99] ve [51, 149] olmak üzere iki farklı aralıkta düzgün dağılmaktadır. Teslim zamanları (d_j) izleyen formül kullanılarak hesaplanmıştır:

$$d_j = \bar{s}_{ij} + \sum_{j=1}^n p_j + u(n-1)\bar{p}$$

burada \bar{s}_{ij} hazırlık sürelerinin ve \bar{p} işlem sürelerinin ortalamasıdır. u , [0, 1] aralığında bir rassal sayıdır. İş sayıları (n) 10, 20 ve 30 olarak alınmış ve her

Tablo 2. F1 Amacı İçin Referans Noktası Olarak Hesaplanan Alt Sınır Değerleri

problem	F1 _{AS}	problem	F1 _{AS}	problem	F1 _{AS}	problem	F1 _{AS}
10_1_1	669	20_1_1	1054	30_1_1	1587	100_2	10183
10_1_2	396	20_1_2	1074	30_1_2	1515	150_2	14904
10_1_3	614	20_1_3	1003	30_1_3	1347	200_2	19195
10_1_4	648	20_1_4	1181	30_1_4	1413	250_2	25200
10_1_5	699	20_1_5	1129	30_1_5	1529		
10_2_1	1174	20_2_1	2102	30_2_1	3214		
10_2_2	986	20_2_2	1956	30_2_2	3053		
10_2_3	965	20_2_3	2075	30_2_3	3221		
10_2_4	956	20_2_4	2297	30_2_4	3152		
10_2_5	1117	20_2_5	2155	30_2_5	3211		

grupta beşer problem türetilmiştir. Test problemleri, sırasıyla iş sayısı, sıra bağımlı hazırlık süresi (eğer [1,99] aralığında düzgün dağılmakta ise 1; [51,149] aralığında düzgün dağılmakta ise 2) ve problem sayısına göre isimlendirilmiştir. Örneğin, 20_2_2 ile gösterilen problem, sıra bağımlı hazırlık süresi [51, 149] aralığında olan 20 işli problemlerden ikinci problemdir. Toplamda 30 problem (1x2x3x5) türetilmiştir. Önerilen yöntemlerin büyük boyutlu problemlerdeki performansını belirlemek amacıyla 100, 150, 200 ve 250 işli dört problem daha türetilmiştir. Büyük problemlerin tümünde sıra bağımlı hazırlık süreleri [51, 149] aralığında düzgün dağılıma uygun olarak türetilmiştir.

5.2 Referans Noktalarının Belirlenmesi

Genişletilmiş ağırlıklı Tchebycheff fonksiyonunda referans noktalar problemlerin her bir amaç için tek amaçlı olarak ele alınıp çözülmesiyle elde edilebilir. Ancak bu çalışmada ele alınan tüm problemlerin tek amaçlı olarak en iyi çözümlerini bulabilmek mümkün olamayacağından her iki amaç için de referans noktası olarak alt sınır değerleri kullanılmıştır.

F2 (ΣT) amacı için referans noktası olarak, tüm problemler için sıfır ($F2_{AS} = 0$) değeri alınmıştır.

F1 (C_{max}) amacı için, referans noktası olarak aşağıda verilen alt sınır formülasyonu kullanılarak hesaplanan değerler Tablo 2'de verilmiştir.

$$AS C_{max} = \text{enk}_j\{h_j\} + \sum_j p_j + \sum_i \text{enk}_j s_{ij}$$

5.3 Önerilen Yöntemlerin Karşılaştırılması

Önerilen sıralama kuralının başarısını test etmek amacıyla WSST, literatürdeki diğer sıralama kuralları EDD, SST ve ATCS ile karşılaştırılmıştır.

Tablo 3. 10, 20 ve 30 İş Problemleri İçin Fayda Fonksiyonu Değerleri ($w_1=0,25$, $w_2=0,75$)

Problem	EDD	SST	ATCS	WSST
10_1_1	279,25	192,75	197,50	281,75
10_1_2	300,25	142,00	142,00	142,00
10_1_3	408,75	194,25	173,50	194,25
10_1_4	449,00	198,25	183,25	198,25
10_1_5	225,25	287,00	224,25	189,75
10_2_1	1245,00	662,00	858,50	757,00
10_2_2	1322,75	681,25	682,75	702,50
10_2_3	2622,25	1261,25	1333,00	1284,50
10_2_4	1728,25	759,75	695,00	790,25
10_2_5	830,50	476,25	513,25	476,25
20_1_1	1784,75	300,00	318,75	300,00
20_1_2	736,25	306,25	320,50	315,00
20_1_3	493,50	293,75	298,50	289,00
20_1_4	486,00	400,75	319,50	333,00
20_1_5	903,75	361,50	345,00	345,00
20_2_1	3272,25	2116,75	1355,75	1832,75
20_2_2	6964,50	2208,50	1999,25	1805,25
20_2_3	6544,50	3715,25	2164,50	2264,50
20_2_4	4460,50	2197,50	1378,25	1287,75
20_2_5	5589,00	2467,75	1490,75	1393,25
30_1_1	1066,25	445,50	446,75	453,50
30_1_2	680,50	436,75	468,75	451,25
30_1_3	3660,75	394,25	409,00	376,75
30_1_4	951,25	391,50	441,00	456,75
30_1_5	1455,50	445,00	452,75	458,25
30_2_1	7865,00	2925,75	1611,50	2042,25
30_2_2	12077,00	4416,00	2920,25	3658,75
30_2_3	8997,25	3297,25	1971,00	1449,00
30_2_4	7401,50	2132,75	1881,75	1982,50
30_2_5	5723,00	4065,25	1622,00	1413,50

Tablo 4. Büyük Boyutlu Problemler İçin Fayda Fonksiyonu Değerleri ($w_1=0,25$, $w_2=0,75$)

Problem	EDD	SST	ATCS	WSST
100_2	106718,75	40299,00	13366,25	10030,50
150_2	215019,00	93667,25	33718,75	17585,25
200_2	462661,00	163534,25	90063,75	45013,00
250_2	610828,50	189097,00	76867,50	26746,50

Problem iki amaçlı olduğu için farklı yöntemlerle elde edilen çözümlerin karşılaştırılmasında toplamsal fayda fonksiyonu kullanılmıştır. Toplamsal fayda fonksiyonu $U=w_1F1 + w_2F2$ olarak tanımlanmıştır. ($w_1=0,25$, $w_2=0,75$) için elde edilen fayda fonksiyonu değerleri Tablo 3 ve 4'te verilmiştir. Tablo 3'te görüldüğü gibi EDD ile hiç başarılı çözüm elde edilemezken, SST, ATCS ve WSST yaklaşık olarak aynı performansı göstermişlerdir. Fayda fonksiyonu göz önünde bulundurulduğunda büyük boyutlu problemlerinin tümünde WSST yöntemi diğer yöntemlerden daha başarılı sonuçlar üretmiştir.

Önerilen WSST sıralama kuralının özellikle büyük boyutlu problemlerdeki başarısı, oldukça ümit vericidir. Bu sonuçların daha da iyileştirilmesi amacıyla önerilen i-WSST ve WSST'nin karşılaştırma sonuçları Tablo 5'te verilmiştir. Tablodan görüldüğü üzere WSST ile 2-opt arama algoritmasının birleştirilmesi, fayda fonksiyonu değerlerinde ($w_1=0,25$, $w_2=0,75$) iken ortalama %5,55, ($w_1=0,75$, $w_2=0,25$) iken ortalama %4,49 oranında iyileştirme sağlamıştır.

i-WSST yönteminin sonuçları ayrıca literatürdeki diğer sıralama kuralları EDD, SST ve ATCS ile de karşılaştırılmıştır. ($w_1=0,25$, $w_2=0,75$) ve ($w_1=0,75$, $w_2=0,25$) için elde edilen fayda fonksiyonu değerleri Tablo 6 ve 7'de verilmiştir. Tablo 6'da görüldüğü gibi ($w_1=0,25$, $w_2=0,75$) iken i-WSST, problemlerin yaklaşık %65'inde daha başarılı olmuştur. ($w_1=0,75$, $w_2=0,25$) durumunda ise bu oran %57'dir. Fayda fonksiyonu dikkate alındığında, büyük boyutlu problemlerin tamamında i-WSST diğer sıralama kurallarından daha başarılı sonuçlar üretmiştir. Özellikle 250

boyutlu test problemindeki iyileştirme oranı dikkat çekicidir. i-WSST ATCS'den ($w_1=0,25$, $w_2=0,75$) iken %67, ($w_1=0,75$, $w_2=0,25$) iken %42 oranında daha başarılı sonuçlar üretmiştir.

Tablo 5. WSST ve i-WSST Algoritmalarının Karşılaştırılması

Problem	$(w_1, w_2) = (0.25, 0.75)$		$(w_1, w_2) = (0.75, 0.25)$	
	WSST	i-WSST	WSST	i-WSST
10_1_1	281,75	221,50	647,25	596,25
10_1_2	142,00	122,75	412,00	368,25
10_1_3	194,25	176,75	582,75	530,25
10_1_4	198,25	188,00	594,75	564,00
10_1_5	189,75	189,75	569,25	569,25
10_2_1	757,00	698,25	1165,00	1116,75
10_2_2	702,50	660,25	959,50	938,75
10_2_3	1284,50	1184,50	1171,50	1117,50
10_2_4	790,25	715,50	1016,75	960,50
10_2_5	476,25	476,25	924,75	924,75
20_1_1	300,00	300,00	900,00	900,00
20_1_2	315,00	302,00	945,00	906,00
20_1_3	289,00	283,25	867,00	849,75
20_1_4	333,00	332,75	999,00	998,25
20_1_5	345,00	345,00	953,00	953,00
20_2_1	1832,75	1631,25	2212,25	2091,75
20_2_2	1805,25	1724,75	2047,75	2052,25
20_2_3	2264,50	2075,00	2273,50	2241,00
20_2_4	1287,75	1151,75	2053,25	2077,25
20_2_5	1393,25	1344,00	2051,75	1984,00
30_1_1	453,50	434,75	1360,50	1304,25
30_1_2	451,25	443,75	1353,75	1331,25
30_1_3	376,75	376,75	1130,25	1130,25
30_1_4	456,75	414,25	1276,25	1242,75
30_1_5	458,25	444,25	1374,75	1332,75
30_2_1	2042,25	1838,25	2992,75	2972,75
30_2_2	3658,75	3541,50	3500,25	3432,50
30_2_3	1449,00	1449,00	2823,00	2815,75
30_2_4	1982,50	1791,25	3019,50	2939,75
30_2_5	1413,50	1239,00	2800,50	2800,50
100_2	10030,50	9680,25	10525,50	10480,75
150_2	17585,25	16822,75	16263,75	16040,25
200_2	45013,00	44705,25	28323,00	28281,75
250_2	26746,50	26087,50	26297,50	26100,50

Tablo 6. ($w_1=0,25$, $w_2=0,75$) İçin Test Sonuçları

Problem	Sıralama Kuralları			i-WSST
	EDD	SST	ATCS	
10_1_1	279,25	192,75	197,50	221,50
10_1_2	300,25	142,00	142,00	122,75
10_1_3	408,75	194,25	173,50	176,75
10_1_4	449,00	198,25	183,25	188,00
10_1_5	225,25	287,00	224,25	189,75
10_2_1	1245,00	662,00	858,50	698,25
10_2_2	1322,75	681,25	682,75	660,25
10_2_3	2622,25	1261,25	1333,00	1184,50
10_2_4	1728,25	759,75	695,00	715,50
10_2_5	830,50	476,25	513,25	476,25
20_1_1	1784,75	300,00	318,75	300,00
20_1_2	736,25	306,25	320,50	302,00
20_1_3	493,50	293,75	298,50	283,25
20_1_4	486,00	400,75	319,50	332,75
20_1_5	903,75	361,50	345,00	345,00
20_2_1	3272,25	2116,75	1355,75	1631,25
20_2_2	6964,50	2208,50	1999,25	1724,75
20_2_3	6544,50	3715,25	2164,50	2075,00
20_2_4	4460,50	2197,50	1378,25	1151,75
20_2_5	5589,00	2467,75	1490,75	1344,00
30_1_1	1066,25	445,50	446,75	434,75
30_1_2	680,50	436,75	468,75	443,75
30_1_3	3660,75	394,25	409,00	376,75
30_1_4	951,25	391,50	441,00	414,25
30_1_5	1455,50	445,00	452,75	444,25
30_2_1	7865,00	2925,75	1611,50	1838,25
30_2_2	12077,00	4416,00	2920,25	3541,50
30_2_3	8997,25	3297,25	1971,00	1449,00
30_2_4	7401,50	2132,75	1881,75	1791,25
30_2_5	5723,00	4065,25	1622,00	1239,00
100_2	106718,75	40299,00	13366,25	9680,25
150_2	215019,00	93667,25	33718,75	16822,75
200_2	462661,00	163534,25	90063,75	44705,25
250_2	610828,50	189097,00	76867,50	26087,50

Tablo 7. ($w_1=0,75$, $w_2=0,25$) İçin Test Sonuçları

Problem	Sıralama Kuralları			i-WSST
	EDD	SST	ATCS	
10_1_1	837,75	578,25	592,50	596,25
10_1_2	624,75	412,00	412,00	368,25
10_1_3	942,25	582,75	520,50	530,25
10_1_4	867,00	594,75	549,75	564,00
10_1_5	675,75	665,00	626,75	569,25
10_2_1	1451,00	1052,00	1149,50	1116,75
10_2_2	1354,25	959,75	930,25	938,75
10_2_3	1794,75	1147,75	1217,00	1117,50
10_2_4	1518,75	965,25	983,00	960,50
10_2_5	1267,50	924,75	967,75	924,75
20_1_1	2010,25	900,00	956,25	900,00
20_1_2	1674,75	918,75	961,50	906,00
20_1_3	1414,50	881,25	895,50	849,75
20_1_4	1458,00	1068,25	958,50	998,25
20_1_5	1719,25	952,50	1035,00	953,00
20_2_1	2950,75	2268,25	2055,25	2091,75
20_2_2	4417,50	2129,50	2221,75	2052,25
20_2_3	4229,50	2737,75	2247,50	2241,00
20_2_4	3667,50	2364,50	2152,75	2077,25
20_2_5	4003,00	2373,25	2074,25	1984,00
30_1_1	2386,75	1336,50	1340,25	1304,25
30_1_2	2041,50	1310,25	1406,25	1331,25
30_1_3	3320,25	1182,75	1227,00	1130,25
30_1_4	2179,75	1174,50	1323,00	1242,75
30_1_5	2570,50	1335,00	1358,25	1332,75
30_2_1	5719,00	3273,25	2948,50	2972,75
30_2_2	7141,00	3644,00	3280,75	3432,50
30_2_3	6051,75	3393,75	3021,00	2815,75
30_2_4	5464,50	2896,25	3015,25	2939,75
30_2_5	4945,00	3645,75	2984,00	2800,50
100_2	45840,25	20413,00	12020,75	10480,75
150_2	86297,00	41435,75	22524,25	16040,25
200_2	173441,00	67516,75	44563,25	28281,75
250_2	228679,50	80035,00	44576,50	26100,50

Ayrıca, yöntemlerin türettikleri çözümler baskınlık açısından da incelenmiştir. Bu amaçla hazırlanan Tablo 8’de her bir yöntemle elde edilen amaç fonksiyonu değerleri (F1 ve F2) ayrı ayrı verilmiştir. i-WSST yönteminde amaçlar genişletilmiş ağırlıklı Tchebycheff fonksiyonu kullanılarak birleştirildiğinden, farklı ağırlıklar için farklı çözümler elde edilebilmektedir. Tablo 8’in son sütununda 11 farklı ağırlık değeri ($(w_1, w_2) \in \{(1;0), (0,9;0,1), (0,8;0,2), (0,7;0,3), (0,6;0,4), (0,5;0,5), (0,4;0,6), (0,3;0,7), (0,2;0,8), (0,1;0,9), (1;0)\}$) ile elde edilen çözümler sunulmuştur. Tabloda hakim (birbirine baskın olmayan çözümler) koyu ve altı çizili olarak işaretlenmiştir.

Tablo 8 elde edilen hakim çözüm sayısı açısından

incelendiğinde, öne çıkan iki çözüm yönteminin i-WSST ve SST olduğu görülmektedir. Ancak SST sıralama kuralı tarafından türetilen çözümler incelendiğinde, F1 amacı için i-WSST’den az bir farkla iyi değerler elde edilmiş olduğu görülürken, yöntem tek bir amacı dikkate aldığından F2 amacı için oldukça kötü değerlerin elde edildiği gözlenmektedir. i-WSST algoritması hem her iki amaç için de makul değerler türetebilmekte hem de farklı ağırlık değerleri için farklı hakim çözümler üretilmesine olanak tanımaktadır. Tablo 8’de sunulan çözümler sadece 11 elemanlı küçük bir ağırlık seti için elde edilmiştir. Bu setin büyütülmesiyle farklı hakim çözümlere erişmek de mümkündür.

Tablo 8. F1 ve F2 Amaç Fonksiyonu Değerleri

Problem	EDD		SST		ATCS		WSST		i-WSST					
	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2
10_1_1	1117	0	<u>771</u>	<u>0</u>	790	0	830	99	806	0	787	33	779	48
10_1_2	787	138	547	7	547	7	547	7	<u>491</u>	<u>0</u>				
10_1_3	1209	142	777	0	<u>694</u>	<u>0</u>	777	0	707	0				
10_1_4	1076	240	793	0	<u>733</u>	<u>0</u>	793	0	752	0				
10_1_5	901	0	854	98	828	23	<u>759</u>	<u>0</u>	<u>759</u>	<u>0</u>				
10_2_1	1554	1142	<u>1247</u>	<u>467</u>	1295	713	1369	553	1326	489	1301	573		
10_2_2	1370	1307	1099	542	<u>1054</u>	<u>559</u>	1088	574	<u>1078</u>	<u>521</u>				
10_2_3	1381	3036	1091	1318	1159	1391	1115	1341	<u>1084</u>	<u>1218</u>				
10_2_4	1414	1833	1068	657	<u>1127</u>	<u>551</u>	1130	677	<u>1083</u>	<u>593</u>	<u>1064</u>	<u>622</u>		
10_2_5	1486	612	<u>1149</u>	<u>252</u>	1195	286	<u>1149</u>	<u>252</u>	<u>1149</u>	<u>252</u>				
20_1_1	2123	1672	<u>1200</u>	<u>0</u>	1275	0	<u>1200</u>	<u>0</u>	<u>1200</u>	<u>0</u>				
20_1_2	2144	267	1225	0	1282	0	1260	0	<u>1208</u>	<u>0</u>				
20_1_3	1875	33	1175	0	1194	0	1156	0	<u>1133</u>	<u>0</u>				
20_1_4	1944	0	1402	67	<u>1278</u>	<u>0</u>	1332	0	1331	0				
20_1_5	2127	496	<u>1248</u>	<u>66</u>	<u>1380</u>	<u>0</u>	<u>1257</u>	<u>41</u>	<u>1257</u>	<u>41</u>	<u>1312</u>	<u>24</u>		
20_2_1	2790	3433	2344	2041	<u>2405</u>	<u>1006</u>	2402	1643	<u>2322</u>	<u>1401</u>				
20_2_2	3144	8238	<u>2090</u>	<u>2248</u>	2333	1888	<u>2169</u>	<u>1684</u>	<u>2216</u>	<u>1561</u>	<u>2169</u>	<u>1684</u>	<u>2149</u>	<u>2156</u>
20_2_3	3072	7702	<u>2249</u>	<u>4204</u>	<u>2289</u>	<u>2123</u>	<u>2278</u>	<u>2260</u>	<u>2324</u>	<u>1992</u>	<u>2283</u>	<u>2224</u>	<u>2250</u>	<u>2332</u>
20_2_4	3271	4857	2448	2114	2540	991	<u>2436</u>	<u>905</u>	<u>2540</u>	<u>689</u>	<u>2496</u>	<u>757</u>	<u>2436</u>	<u>905</u>
20_2_5	3210	6382	2326	2515	2366	1199	2381	1064	<u>2304</u>	<u>1024</u>				
30_1_1	3047	406	1782	0	1787	0	1814	0	<u>1739</u>	<u>0</u>				
30_1_2	2722	0	<u>1747</u>	<u>0</u>	1875	0	1805	0	1775	0				
30_1_3	3150	3831	1577	0	1636	0	<u>1507</u>	<u>0</u>	<u>1507</u>	<u>0</u>				
30_1_4	2794	337	<u>1566</u>	<u>0</u>	1764	0	1686	47	1657	0				
30_1_5	3128	898	1780	0	1811	0	1833	0	<u>1777</u>	<u>0</u>				
30_2_1	4646	8938	3447	2752	<u>3617</u>	<u>943</u>	3468	1567	<u>3540</u>	<u>1271</u>	<u>3424</u>	<u>1356</u>		
30_2_2	4673	14545	<u>3258</u>	<u>4802</u>	<u>3461</u>	<u>2740</u>	3421	3738	<u>3378</u>	<u>3596</u>				
30_2_3	4579	10470	3442	3249	3546	1446	<u>3510</u>	<u>762</u>	<u>3510</u>	<u>762</u>	<u>3475</u>	<u>838</u>	<u>3429</u>	<u>1357</u>
30_2_4	4496	8370	<u>3278</u>	<u>1751</u>	3582	1315	3538	1464	<u>3514</u>	<u>1217</u>	3448	2729		
30_2_5	4556	6112	<u>3436</u>	<u>4275</u>	3665	941	<u>3494</u>	<u>720</u>	<u>3534</u>	<u>474</u>	<u>3494</u>	<u>720</u>	<u>3465</u>	<u>987</u>
100_2	15401	137158	<u>10470</u>	<u>50242</u>	11348	14039	<u>10773</u>	<u>9783</u>	<u>10881</u>	<u>9280</u>	<u>10676</u>	<u>10021</u>		
150_2	21936	279380	<u>15320</u>	<u>119783</u>	16927	39316	<u>15603</u>	<u>18246</u>	<u>15649</u>	<u>17214</u>	<u>15580</u>	<u>20409</u>		
200_2	28831	607271	<u>19508</u>	<u>211543</u>	21813	112814	19978	53358	<u>20070</u>	<u>52917</u>	<u>19844</u>	<u>53129</u>		
250_2	37605	801903	<u>25504</u>	<u>243628</u>	28431	93013	<u>26073</u>	<u>26971</u>	<u>26107</u>	<u>26081</u>				

6. SONUÇLAR

Bu çalışmada sıra bağımlı hazırlık süreli, son işin tamamlanma zamanını (C_{max}) ve toplam gecikmeyi (ΣT) enküçükmeyi amaçlayan tek makine çizelgeleme problemi ele alınmıştır. NP-zor sınıfında yer alan bu iki ölçütlü problemin çözümü için iki farklı çözüm yöntemi önerilmiştir. Bunlardan ilki, SST sıralama kuralının ağırlıklandırılmış hali olan WSST, diğeri ise başlangıç sıralamasını WSST kuralına göre yapan ve sonrasında bu sıralamayı yerel aramayla (2-opt) iyileştiren i-WSST algoritmasıdır. Önerilen çözüm yöntemlerinin performansı rassal türetilen test problemleri kullanılarak hem fayda fonksiyonu

değeri hem de hakim çözüm sayısı açısından karşılaştırılmıştır. Fayda fonksiyonu değerleri dikkate alındığında, i-WSST algoritması hem WSST hem de literatürde yer alan SST, EDD ve ATCS sıralama kurallarından daha iyi performans göstermiştir. Gerek WSST gerekse i-WSST büyük boyutlu problemlerde karşılaştırma yapılan sıralama kurallarının çok üzerinde performanslar göstermişlerdir. Hakim çözüm sayısı dikkate alındığında, i-WSST öne çıkmaktadır. Ayrıca, i-WSST algoritması hem her iki amaç için de makul değerler türetebilmekte hem de farklı ağırlık değerleri için farklı hakim çözümler üretebilmesine olanak tanımaktadır.

KAYNAKÇA

1. Armentano, V.A., Mazzini, R. 2000. "A Genetic Algorithm For Scheduling On A Single Machine With Set-Up Times And Due Dates," *Production Planning and Control*, 11, 713–720.
2. Asano, M., Ohta, H. 1999. "Scheduling With Shutdowns And Sequence Dependent Set-Up Times," *International Journal of Production Research*, 37, 1661–1676.
3. Chang, T.Y., Chou, F.D., Lee, C.E. 2004. "A Heuristic Algorithm To Minimize Total Weighted Tardiness On A Single Machine With Release Dates And Sequence-Dependent Setup Times," *Journal of the Chinese Institute of Industrial Engineers*, 21, 289–300.
4. Chankong, V., Haimes, Y.Y. 1983. "Multiobjective Decision Making: Theory and Methodology", Elsevier Science Publishing Co, New York.
5. Choobineh, F.F., Mohebbi, E., Khoo, H. 2006. "A Multi-Objective Tabu Search For A Single-Machine Scheduling Problem With Sequence-Dependent Setup Times," *European Journal of Operational Research*, 175, 318–337.
6. Ehrgott, M. 2005. "Multicriteria Optimization," Springer-Verlag, 2nd edition.
7. Eren, T., Guner, E. 2006. "A Bicriteria Scheduling With Sequence Dependent Setup Times," *Applied Mathematics and Computation*, 179, 378–385.
8. Franca, P.M., Mendes, A., Moscato, P. 2001. "A Memetic Algorithm For The Total Tardiness Single Machine Scheduling Problem," *European Journal of Operational Research*, 132, 224–242.
9. Gagne, C., Price, W.L., Gravel, M. 2002. "Comparing an ACO Algorithm With Other Heuristics For The Single Machine Scheduling Problem With Sequence-Dependent Setup Times," *Journal of the Operational Research Society*, 53, 895–906.
10. Gupta, S.R., Smith, J.S. 2006. "Algorithms For Single Machine Total Tardiness Scheduling With Sequence Dependent Setups," *European Journal of Operational Research*, 175, 722–739.
11. Kolahan, F., Liang, M. 1998. "An adaptive TS Approach To JIT Sequencing With Variable Processing Times And Sequencedependent Setups," *European Journal of Operational Research*, 109, 142–159.
12. Lee, S.M., Asllani, A.A. 2004. "Job Scheduling With Dual Criteria And Sequence-Dependent Setups: Mathematical Versus Genetic Programming," *Omega*, 32, 145–153.
13. Luc, D.T. 1989. "Theory of Vector Optimization," *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin.
14. Mendes, A.S., Franca, P.M., Moscato, P. 2002. "Fitness Landscapes For The Total Tardiness Single Machine Scheduling Problem," *Neural Network World*, 12, 165–180.
15. Miller, D.M., Chen, H.C., Matson, J., Liu, Q. 1999. "A Hybrid Genetic Algorithm For The Single Machine Scheduling Problem," *Journal of Heuristics*, 5, 437–454.
16. Pinedo, M. 2002. "Scheduling: Theory, Algorithms and Systems," 2nd Edition, Prentice Hall.
17. Rabadi, G., Mollaghasemi, M., Anagnostopoulos, G.C. 2004. "A Branch-And-Bound Algorithm For The Early/Tardy Machine Scheduling Problem With A Common Due-Date And Sequence Dependent Setup Time," *Computers and Operations Research*, 31, 1727–1751.
18. Rajendran, C., Ziegler, H. 2003. "Scheduling to Minimize the Sum of Weighted Flowtime and Weighted Tardiness of Jobs in A Flowshop with Sequence-Dependent Setup Times," *European Journal of Operational Research*, 149 (3), 513–522.
19. Shin, H.J., Kim, C.O., Kim, S.S. 2002. "A Tabu Search Algorithm For Single Machine Scheduling With Release Times, Due Dates, And Sequence-Dependent Set-Up Times," *International Journal of Advanced Manufacturing Technology*, 19, 859–866.
20. Tan, K.C., Narasimhan, R. 1997. "Minimizing Tardiness On A Single Processor With Sequence-Dependent Setup Times: A Simulated Annealing Approach," *Omega*, 25, 619–634.
21. Tan, K.C., Narasimhan, R., Rubin, P.A., Ragatz, G.L. 2000. "A Comparison Of Four Methods For Minimizing Total Tardiness On A Single Processor With Sequence Dependent Setup Times," *Omega*, 28, 313–326.
22. Wang, L., Wang, M. 1997. "Hybrid Algorithm For Earliness-Tardiness Scheduling Problem With Sequence Dependent Setup Time," *Proceedings of the IEEE Conference on Decision and Control*, 2, 1219–1222.