

Tekrarlı Açgözlü Algoritması ile Esnek Atölye Tipi Çizelgeleme Problemi Üzerine Bir Uygulama

Yunus DEMİR^{1*}

ÖZET: En iyileme (optimizasyon), belirli kriterler çerçevesinde muhtemel çözümler arasından en iyinin (min/max) seçilmesidir. En iyileme problemlerinin çözümü için; kesin çözüm yöntemleri, yakınsama metotları, meta-sezgisel teknikler gibi farklı sınıflarda birçok yaklaşım geliştirilmiştir. Ancak gerçek hayat problemlerinin devasa boyutlara ulaşması, araştırmacıları kısa zamanda, kabul edilebilir çözümler veren meta-sezgisel tekniklere yöneltmiştir. Bu çalışma ile meta-sezgisel algoritmaların çeşitli alanlarda uygulanması konusunda kısıtlı olan Türkçe literatüre katkı sağlanması amaçlanmıştır. Bu doğrultuda, sade yapısı ile ön plana çıkan tekrarlı açgözlü algoritması ile bir uygulama yapılmıştır. Uygulama için esnek atölye tipi çizelgeleme problemi ele alınmıştır. Bu çalışmada, yapım-yıkım fazında probleme özgü kritik yol tabanlı bir yaklaşım geliştirilmiştir. Ayrıca iterasyon sayısına bağlı olarak azalan kalitede çözümlerin kabulüne dayalı özgün bir yaklaşım önerilmiştir. Geliştirilen algoritmanın performansı, Fattahi ve ark., (2007) tarafından geliştirilen örnek problemler ile test edilmiş ve sonuçlar literatürde yapılan diğer çalışmalar ile karşılaştırılmıştır.

Anahtar Kelimeler: Kombinatoriyal en iyileme, tekrarlı açgözlü algoritma, esnek atölye tipi çizelgeleme problemi

An Application on Flexible Job Shop Scheduling Problem with Iterated Greedy Algorithm

ABSTRACT: Optimization is the selection of the best possible solution (min / max) under certain criteria. For the solution of optimization problems; many approaches have been developed in different classes such as exact solution methods, approximation methods, meta-heuristic techniques. However, the enormous size of real-life problems have led researchers to meta-heuristic techniques that provide acceptable solutions in a short time. In this study, an application has been made with the iterated greedy algorithm, which stands out with its simple structure. Flexible job shop scheduling problem is addressed for application. Basically, the iterative greedy algorithm starts the optimization process with a single solution. The current solution then enters the construction-destruction phase to find a better solution. The solution obtained after this stage is replaced with the incumbent solution according to the previously determined acceptance criteria. This cycle, which consists of two operators, construction and destruction, continues until a certain stopping criterion is met. In this study, a problem-specific critical path-based approach was developed in the construction-destruction phase. In addition, a novel approach based on the acceptance of solutions of decreasing quality depending on the number of iterations is proposed. The aim of this study is to contribute to the limited Turkish literature on the application of meta-heuristic algorithms in various fields.

Keywords: Combinatorial optimization, Iterated greedy algorithm, Flexible job shop scheduling problem

¹Yunus DEMİR (Orcid ID: 0000-0003-3868-1860), Bursa Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Endüstri Mühendisliği Bölümü, Bursa, Türkiye

*Sorumlu Yazar/Corresponding Author: Yunus DEMİR, e-mail: yunus.demir@btu.edu.tr

GİRİŞ

Çizelgeleme faaliyetleri, üretim ve hizmet sektörü için kritik karar süreçlerindedir. Sevkiyat, üretim planlama, dağıtım, ulaştırma, bilgi işleme gibi birçok alanda kullanılan çizelgeleme iyi yapıldığında işletme kaynaklarının verimli kullanımını beraberinde getirir (Antczak ve ark., 2009). Bu çalışmada klasik atölye tipi çizelgeleme (AÇP, Job Shop Scheduling Problem-JSP) probleminin uzantısı olan esnek atölye tipi çizelgeleme problemi (EAÇP, Flexible Job Shop Scheduling Problem-FJSP) ele alınmıştır. AÇP’de önceden belirlenmiş sırada ve yine önceden belirlenmiş makinelerde, işlenmeyi bekleyen operasyonlar çizelgelenir. Ancak EAÇP’de operasyonlar birden fazla makinede işlem görebilir. Dolayısıyla EAÇP, işlem için alternatifler arasından makinenin seçilmesi ve makineler üzerindeki operasyonların belirli amaçları en iyilemek üzere sıralanması olmak üzere iki alt problemden oluşur. Bu haliyle EAÇP klasik atölye tipi çizelgeleme probleminin özelleşmiş halidir. Klasik atölye tipi çizelgeleme problemi NP-hard zorlukta olduğu daha önceden Garey ve Johnson (1979) tarafından ispatlanmıştır. Dolayısıyla bir tür sıralama problemin olan AÇP’ye ilaveten atama problemini de içeren EAÇP de NP-hard zorlukta bir problem tipidir (Gao ve ark., 2008).

Ele alınan problem şu şekilde formülize edilmektedir: m makinede $\mathcal{M} = \{M_1, M_2, \dots, M_m\}$ işlenmeyi bekleyen n adet iş $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$ bulunmaktadır. Her J_i işi h_i operasyondan oluşmaktadır. Her operasyon O_{ij} (i işinin j operasyonunu) birden fazla farklı makinede $\mathcal{M}_k \subset \mathcal{M}$ işlenebilir. O_{ij} operasyonunun alternatif makineler üzerindeki işlem süresi p_{ijk} şeklinde ifade edilir. Problemde amaç toplam tamamlanma zamanının C_{max} ($\max_{i=1,2,\dots,n} C_i$) minimizasyonudur. Problem ile ilgili genel varsayımlar şu şekildedir:

- Her makine aynı anda sadece bir işi işleyebilir.
- Operasyonlar kesintisiz bir şekilde yapılır.
- Bir operasyon aynı anda birden fazla makinede işlenemez.
- Hazırlık ve taşıma zamanları ihmal edilmiştir.

EAÇP ile ilgili ilk çalışma Bruker ve Schlie (1990) tarafından yapılmıştır. Önemli bir yöneylem araştırması problemi olan EAÇP, doksanlardan günümüze kadar popüleritesini korumakta ve araştırmacılar tarafından yoğun bir şekilde çalışılmaktadır. Problemin en yalın hali (C_{max} minimizasyonu) için birçok farklı algoritma geliştirilmiş, yanı sıra onlarca farklı versiyonu için çalışmalar yapılmıştır. EAÇP ile ilgili kapsamlı literatüre Chaudhry ve Khan (2016), Amjad ve ark. (2018), Gao ve ark. (2019), Xie ve ark. (2019), Demir ve İşleyen (2014)’ün çalışmalarından ulaşılabilmektedir. Chaudhry ve Khan (2016) 1990-2014 yılları arasında yapılmış 404 yayını taramıştır. Yazarlar EAÇP’nin 55 farklı amaç için çözüldüğünü ortaya koymuşlardır. Amjad ve ark. (2018) EAÇP’nin çözümü için geliştirilmiş genetik algoritma tabanlı çözüm yaklaşımlarını incelemiştir. Gao ve ark. (2019) EAÇP için geliştirilen sürü zekâsı (*swarm intelligence*) ve evrimsel algoritmalar (*evolutionary algorithms*) taramıştır. Xie ve ark. (2019) EAÇP literatürünü çözüm metotlarına göre kesin çözüm, sezgisel ve meta-sezgisel yaklaşımlar olmak üzere üçe ayırarak incelemiştir. Ayrıca yazarlar çalışmalarında EAÇP ile ilgili gerçek hayat uygulamalarını da tanıtmış ve gelecek araştırma fırsatları ile ilgili bir değerlendirmeye yer verilmiştir. Demir ve İşleyen (2014) EAÇP için geliştirilmiş matematiksel model yaklaşımlarını sınıflandırmış ve performanslarını karşılaştırmalı bir şekilde sunmuşlardır.

Bu çalışmada klasik EAÇP problemi için tekrarlı açgözlü algoritması (TAA, Iterated Greedy Algorithm-IGA) önerilmiştir. Araştırmalarımıza göre literatürde EAÇP problemi için TAA’ya iki defa başvurulmuştur. Bunlardan ilki Aqel ve ark. (2018) tarafından yapılmış çalışmadır. Yazarlar çalışmalarında, tek çözüm tabanlı yapıda olan TAA’yı popülasyon tabanlı olacak şekilde modifiye

etmişlerdir. Aynı yazar grubu tarafından gerçekleştirilen bir diğer çalışmada klasik TAA'yı ardışık bir şekilde işleyen iki fazlı bir şekilde düzenlemişlerdir (Aqel ve ark., 2019). EAÇP, atama ve sıralama olmak üzere iki alt problemden oluşması bu konuda üretilen çözüm yöntemlerini iki fazlı yaklaşımlar geliştirmeye itmiştir. Bu çalışmada da benzer bir yaklaşım takip edilmiş atama ve sıralama alt problemleri için farklı stratejiler takip edilmiştir. Yıkım-yeniden (*destruction-reconstruction*) yapım fazı için kritik yol temelli yeni bir yaklaşım geliştirilmiştir. Ayrıca yeni bir kabul yapısı oluşturulmuştur.

Çalışmanın devam eden kısımları şu şekilde organize edilmiştir: Materyal ve yöntem bölümünde TAA'nın temel yapısı ve önerilen TAA'dan bahsedilmiş, kullanılan/geliştirilen operatörlere geniş yer verilmiştir. Bulgular ve tartışma bölümünde geliştirilen algoritma test edilmiş ve elde edilen sonuçlar literatürde daha önce yapılmış çalışmalar ile karşılaştırılmıştır. Sonuç bölümünde çalışma ile ilgili genel bir değerlendirmeye yer verilmiştir.

MATERYAL VE METOT

Önerilen Çözüm Yaklaşımı: Tekrarlı Aç Gözlü Algoritması

Yapıcı sezgiseller ile yerel arama tekniklerinin birleşiminden oluşan tekrarlı açgözlü algoritması (TAA) ilk defa Jacobs ve Brusco (1995) tarafından önerilmiş, çizelgeleme problemlerinde ilk defa Ruiz ve Stütze (2007) tarafından kullanılmıştır. TAA, temel olarak yapıcı bir sezgisel (*constructive heuristics*) ile başlangıç çözümünün oluşturulması ile başlar. Daha sonra önceden tanımlanmış durdurma kriteri sağlanana kadar tekrarlı bir şekilde işletilen iki ayrı fazdan oluşmaktadır. Yıkım (*destruction*) fazı olarak isimlendirilen ilk fazda çözüme ait bazı bileşenler çıkarılır ve yeniden yapım (*reconstruction*) olarak isimlendirilen ikinci fazda, çıkarılan bileşenler probleme özgü aç gözlü teknikler ile yeniden çözüme ilave edilir. İsteğe bağlı olarak çözüm kalitesini artırmak adına başlangıç çözüm ve yeni oluşturulmuş çözüm (*reconstructed solution*) sonrası yerel arama uygulanır. Daha sonra yeni oluşan çözümün (*reconstructed solution*) mevcut çözüm (*incumbent solution*) ile değiştirilip değiştirilmeyeceğine karar vermek için bir karar kriteri kullanılır (Ruiz ve Stütze, 2007). Bu çalışmada ele alınan problemin çözümü için geliştirilen TAA alt başlıklar halinde tanıtılmış olup algoritmanın genel yapısı Şekil 1'de sunulmuştur.

Prosedür: Tekrarlı Açgözlü Algoritması

S_0 := Başlangıç çözümünün oluşturulması

S_i := Yerel arama(S_0)

% Seçmeli

While (Sonlandırma kriteri karşılanmadığı sürece)

S_D := Yıkım(S_i)

S' := Yeniden yapım(S_D, S_R)

S_C := Yerel arama(S')

% Seçmeli

S_i := Karar kriteri(S_C, S_i)

endWhile

return $S_{max/min}$

Şekil 1. TAA Genel Yapısına Ait Pseudo (Sözde) Kodu

Ele alınan problemde C_{max} minimizasyonu yapılmaktadır. Yani Gantt diyagramı üzerinde tüm operasyonlar mümkün olduğunca sola (sıfır noktasına) yaslanmaya çalışılmaktadır. Dolayısıyla optimum çözümün, makine iş yükleri dengeli dağıtılmış çizelgelerde aranması makul bir yaklaşım olacaktır. Bu bağlamda başlangıç çözüm ve yeniden yapım aşamasında makine atama alt problemi için

Kacem ve ark. (2002) tarafından geliştirilen, güncel makine iş yüklerini dikkate alan yapıcı bir sezgisel benimsenmiştir. EAÇP iki alt problemden oluştuğundan bu çalışmada makine atama ve operasyon sıralama olmak üzere iki farklı dizi halinde gösterilmiş ve önerilen TAA operatörleri atama ve sıralama alt problemleri için farklı tasarlanmıştır. Klasik TAA'da yıkım-yeniden yapım aşamasında d kadar bileşen çıkarılarak dizideki diğer pozisyonlara yerleştirilir. Permutasyon gösterimine uygun olan operasyon sıralama kısmı için bu yapı sorunsuz sonuç verir. Ancak makine atama kısmında uygunsuz çözümlere sebep olabilir. Bu yüzden makine atama kısmı için özgün bir yıkım-yeniden yapım operatörü geliştirilmiştir. Geliştirilen operatörün işleyişine yönelik detaylar ilgili bölümde verilmiştir. Ayrıca her iterasyonda azalan çözüm kalitesi tabanlı yeni bir kabul kriteri oluşturulmuştur. İlk iterasyonda en az ulaşılan en iyi sonuçtan daha iyi bir sonuç aranır. Her iterasyonda çözüm kalitesi % q kadar düşürülmektedir. Önerilen TAA'nın Pseudo (Sözde) Kodu Şekil 2'de verilmiştir.

Prosedür: Önerilen Tekrarlı Açgözlü Algoritması

```

 $\alpha$  ( $\alpha_1 \cup \alpha_2$ ):= Başlangıç çözümün oluşturulması
 $\alpha_i$ := Yerel arama( $\alpha_2$ )           Sadece  $\alpha_r$  için
 $q:=1$ 
While (İterasyon limitine ulaşıncaya kadar)
  do
     $\alpha_d, \alpha_r$ := Yıkım( $\alpha_0$  ( $\alpha_1 \cup \alpha_2$ ))
     $\alpha'$ := Yeniden yapım( $\alpha_d, \alpha_r$ )
     $\alpha_c$ := Yerel arama( $\alpha'$ )           Sadece  $\alpha_r$  için
     $q:= q-0.05$ 
    while( $\alpha_i < [\alpha_c * q]$ )
  endWhile
return  $C_{max}$ 

```

Şekil 2. Önerilen TAA'ya Ait Pseudo (Sözde) Kodu

Önerilen TAA'yı daha iyi anlatmak adına Çizelge 1'deki açıklayıcı örnek sunulmuştur. Örnek Fattahi ve ark. (2007) tarafından oluşturulan test problemlerinden MFJS5 kodlu problemdir. 5 makine, 3 iş ve her işe ait 3 operasyonun olduğu bu örnekte her operasyon ikişer alternatif makinede işlem görebilmektedir.

Çizelge 1. Açıklayıcı örnek probleme ait bilgiler

	O1		O2		O3	
	Mak.	İnd. Süre	Mak.	İnd. Süre	Mak.	İnd. Süre
J1	1	117	4	140	4	150
	2	125	2	130	5	160
J2	1	214	3	55	5	78
	3	150	2	66	3	65
J3	1	87	4	70	4	190
	2	62	3	80	5	100

Çözümün Gösterimi

Bu gösterim Gen ve ark. (1997) tarafından geliştirilmiştir. İki ayrı tam sayılı diziden oluşmaktadır. Her dizinin uzunluğu toplam operasyon sayısı kadardır. Gösterimin ilk kısmı (α_1) makine atama, diğer kısmı (α_2) operasyon sıralama için kullanılmaktadır. Gösterimin her iki parçası için aşağıda bir örnek verilmiştir (Şekil 3).

α_1

O11	O12	O13	O21	O22	O23	O31	O32	O33	Operasyon
1	4	5	3	2	5	1	3	4	

 α_2

1	2	3	4	5	6	7	8	9	Okuma sırası
1	2	1	3	2	1	3	2	3	

Şekil 3. Problemin gösterimi

α_1 kısmında her hücre bir operasyonu ifade etmektedir. Hücre içindeki değer ilgili operasyonun atandığı makine indisi göstermektedir. Örneğin 3. hücredeki 5 değeri, 1 numaralı işin 3 numaralı operasyonunun (O_{13}), 5 numaralı makineye atandığını göstermektedir.

α_2 kısmında hücre içerisindeki değerler iş indisini ifade etmektedir. Soldan sağa doğru okunarak operasyonların atandıkları makinelerdeki işleme sırası belirlenir. Karşılaşma sayısı operasyon indisini belirler. Örneğin soldan sağa okunduğunda 3. Hücredeki “1” işine ikinci defa karşılaşılmış olur. Bu 1. işin ikinci operasyonunu (O_{12}) ifade eder. Bu gösterimin permutasyon gösteriminden en önemli farkı, operasyonlar arası bağlılığın gözetilmesi ve her seferinde uygun çözüm elde edilebilmesidir.

Yapıcı sezgisel ve Başlangıç Çözüm Oluşturulması

Bu bölümde başlangıç çözümün oluşturulmasında ve algoritmanın yeniden yapım (*reconstruction*) fazında kullanılan yapım sürecinden bahsedilmiştir. Genel olarak bu süreç parçalı bir çözüme, uygun bir şekilde tanımlanmış çözüm bileşenlerinden oluşan U kümesinden alınan bir bileşenin eklenmesi ile yürütülür (Rodriguez ve ark., 2013). Bu çalışmada kullanılan yapıcı sezgisel, makine atama (α_1) ve operasyon sıralama (α_2) kısımları için iki ayrı prosedür şeklinde tanımlanmıştır.

Makine atama kısmı için Kacem ve ark. (2002) tarafından geliştirilen yaklaşım benimsenmiştir. Bu yaklaşıma göre makine ataması yapılmamış operasyonlar arasından rastgele biri seçilir. Seçilen operasyonun, o an en düşük iş yüküne sahip makineye ataması yapılır ve makine iş yükleri güncellenir. Tüm operasyonların ataması yapılanaya kadar bu döngü devam ettirilir. Konuyla ilgili detaylı açıklamaya Kacem ve ark. (2002), Pezzella ve ark. (2008) ve Zhang ve ark. (2011)’in çalışmalarından ulaşılabilir. Operasyon sıralama kısmı tamamen rassal oluşturulmuştur.

Amaç (Uygunluk) Değerinin Hesaplanması

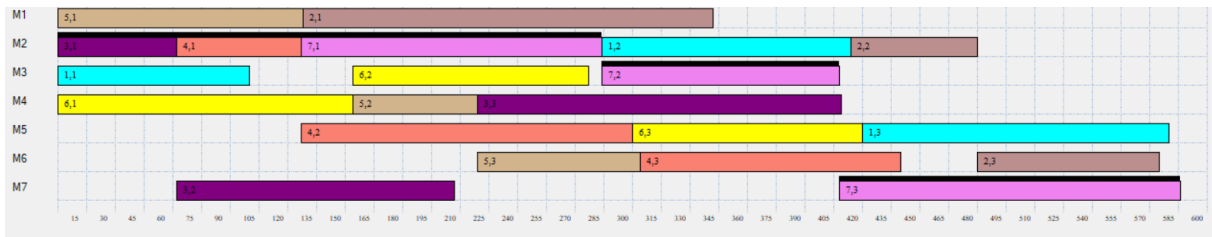
Operasyonlar arasına girebilecek gereksiz boşluklar sebebiyle çözüm gösterimi üzerindeki veri ile sonsuz sayıda çizelge oluşturulabilir (Gao ve ark., 2008). Böylesine büyük bir çözüm uzayında optimumu aramak makul bir yöntem olmayacağından Pinedo (2002) tarafından ortaya konulan aktif çizelge kavramından faydalanılmıştır. Pinedo 2002 çizelgeleri gecikmesiz (*non-delay*), aktif (*active*) ve yarı aktif (*semi-active*) olmak üzere üç sınıfa ayırmıştır. Bu yaklaşıma göre her optimum çizelge aynı zamanda aktif çizelgedir (tersi geçerli değildir). Dolayısıyla çözüm uzayını daraltmak adına bu yaklaşım tercih edilmiştir. Öncelik tabanlı çözümlenme (*Priority-based decoding*) şeklinde de isimlendirilen bu yaklaşım ile ilgili detaylara Gao ve ark. (2008) ve Zang ve ark. (2011) çalışmalarından ulaşılabilir.

Yıkım ve Yeniden Yapım

Bu aşamada n elemandan oluşan mevcut çözümden (*incumbent solution*) d kadar bileşen çıkarılarak parçalı aday bir çözüm α_d (*partial candidate solution*) ve $n-d$ elemandan oluşan çıkarılmış

elemanlar kümesi α_r oluşturulur. Ardından gelen yeniden yapım sürecinde, çıkarılan parçalar yapıcı bir sezgisel ile parçalı aday çözüme sırayla eklenerek tüm bir çözüm ξ_c elde edilir (Lin ve ark., 2013).

Bu çalışmada yıkım ve yeniden yapım süreci, gösterimin makine atama (α_1) ve operasyon sıralama (α_2) kısımları için ayrı ayrı tasarlanmıştır. Makine atama (α_1) kısmı için proje yönetimde kullanılan kritik yol yaklaşımından faydalanılmıştır. Çıkarılacak elemanların bir kısmı (d_a) kritik yol üzerindeki operasyonlardan ve diğer bir kısmı (d_b) kritik olmayan operasyonlardan seçilmektedir. Literatürde çıkarılacak eleman sayısı (d_a+d_b) yıkım derecesi (*degree of destruction*) olarak ifade edilir ve bir problem parametresidir. En erken ve en geç tamamlanma zamanı eşit olan operasyonlar kritik operasyonlardır ve kritik yol C_{max} 'ı belirleyen ardışık operasyonları içermektedir. Örnek olarak Çizelge 1'de sunulan probleme ait optimum çözümün Gantt diyagramı Şekil 4'te sunulmuştur. Her renk bir işi ifade etmektedir. Kutucuklar içerisinde ilgili operasyonun hangi işe ait olduğu ve kaçınıcı sırada geldiği verilmiştir (İş no, operasyon sıra no). Üzeri koyu çizgiler ile belirtilen operasyonlar kritik operasyondur.

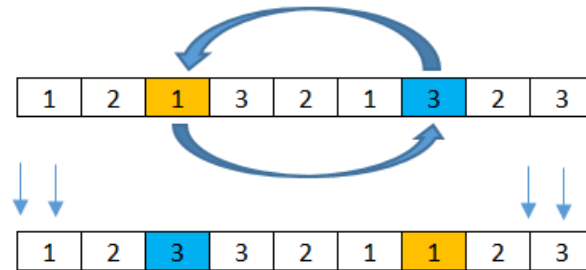


Şekil 4. Gantt diyagramı üzerinde kritik yol gösterimi

Bu çalışmada yıkım aşamasında kritik operasyonlardan $d_a=2$ ve kritik olmayan operasyonlardan $d_b=3$ eleman olmak üzere toplam 5 eleman çıkarılmaktadır. Yeniden yapım aşaması da aynı şekilde makine atama (α_1) ve operasyon sıralama (α_2) kısımları için ayrı ayrı ele alınmıştır.

Makine atama kısmı için önce çıkarılan elemanlar arasından rastgele biri seçilmiştir. Daha sonra seçilen işin yapılabileceği alternatif makinelerden iş yükü en az olan alternatife atanması yapılmıştır. Ancak çözüm çeşitliliğini artırmak adına kritik operasyonların mevcut makineye atanması engellenmiştir. Kritik olmayanlar için böyle bir sınırlandırma yoktur. Yani bu aşamasında, başlangıç çözümde kullanılan yaklaşıma benzer bir yaklaşım kullanılmıştır.

Operasyon sıralama kısmı için Riahi ve ark. (2020) tarafından da kullanılan değiş-tokuş hareketi benimsenmiştir (swap move). Bu yaklaşıma göre her defasında dizi içerisinde aynı olmamak üzere iki pozisyon belirlenmekte ve bu iki pozisyon içerisindeki bilgi birbiri ile yer değiştirmektedir. Belirtilen süreç Şekil 5'te resmedilmiştir.

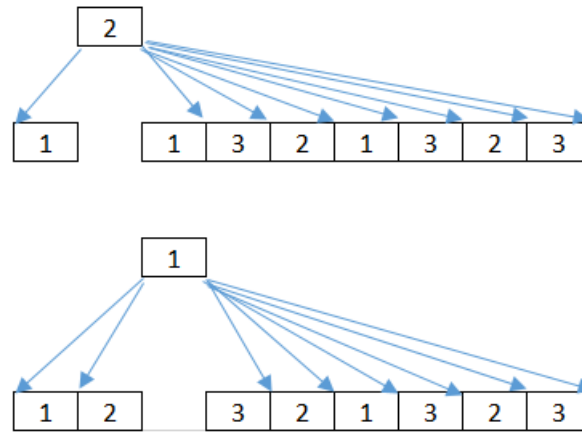


Şekil 5. Operasyon sıralama kısmı için yıkım-yeniden yapım fazı

Buna göre 3 ve 7 numaralı pozisyonlar rastgele seçilmiştir. Daha sonra bu hücrelerdeki değerler yer değiştirilmiştir.

Yerel Arama

Bu çalışmada sadece operasyon sıralama (α_2) kısmı için yeniden yapım fazından sonra ulaşılan sonucu daha da iyileştirmek adına ekleme komşuluk operatörüne dayalı bir yerel arama prosedürü uygulanmıştır. Bu yaklaşım değiş tokuş (swap) operatörü ile literatürde en çok kullanılan komşuluk operatörüdür. Ruiz and Stützle (2007) tarafından oluşturulan ve TAA içinde kullanılan bu yerel arama metodu genel olarak şu şekilde işlemektedir: α_2 dizisi içerisinde s pozisyonunda bulunan iş indeksi çıkarılır. Daha sonra çıkarılan indeks, n boyutlu α_2 dizisi içerisinde çıkarıldığı yer hariç n-1 muhtemel pozisyona eklenir. n-1 pozisyona ekleme sonucu elde edilen sonuçlar arasından en iyisi α_2 çözümü α_2 'den iyi ise yeni çözüm α_2 olur. Bu işlem tüm n pozisyon için tekrarlanır. Süreç Şekil 6'da resmedilmiştir:



Şekil 6. Operasyon sıralama (α_2) kısmı için yeniden yapım fazı

Şekil 6'dan görüleceği üzere ikinci pozisyondan sökülen 2 indeksi ikinci pozisyon harici tüm konumlarda denenmiştir. Daha sonra Üçüncü pozisyondan sökülen 1 indeksi diğer tüm pozisyonlarda denemiştir. Tüm pozisyonlar için aynı süreç tekrarlanır.

BULGULAR VE TARTIŞMA

Önerilen TAA C# dilinde kodlanmış ve 1.8 GHz i5 işlemci ve 8.0 GB RAM özelliklerinde bir bilgisayarda koşulmuştur. Önerilen model literatürdeki diğer bazı çözüm yöntemleri ile Cmax amaç değerlerine göre karşılaştırılmıştır. Karşılaştırma için Fattahi ve ark. (2007) tarafından oluşturulmuş test problemleri kullanılmıştır. Yirmi örnekten oluşan problem seti küçük (SFJS1-SFJS10) ve orta-büyük (MFJS1-MFJS10) olmak üzere iki gruba ayrılmıştır. Test problemlerinin boyutlarına ait bilgiler sırasıyla iş sayısı (i), operasyon sayısı (j) ve makine sayısını (k) göre Çizelge 2'de verilmiştir.

Önerilen TAA'da dört girdi parametresi bulunmaktadır. Bunlar; maksimum iterasyon sayısı (MİS), her iterasyonda indirilecek çözüm kalitesi oranı (q), yıkım aşamasında kritik yol üzerinden çıkarılacak eleman sayısı (d_a) ve kritik olmayan operasyonlardan çıkarılacak eleman sayısıdır (d_b). Deney tasarımı yapılarak aşağıdaki parametreler şu şekilde belirlenmiştir.

Parametre	Değer
MİS	100
q	0.05
d_a	2
d_b	3

Çizelge 2. Test problemlerine ait boyut bilgileri

	i	j	k		i	j	k	
Küçük Boyutlu	SFJS1	2	2	2	MFSJ1	5	3	6
	SFJS2	2	2	2	MFSJ2	5	3	7
	SFJS3	3	2	2	MFSJ3	6	3	7
	SFJS4	3	2	2	MFSJ4	7	3	7
	SFJS5	3	2	2	MFSJ5	7	3	7
	SFJS6	3	3	2	MFSJ6	8	3	7
	SFJS7	3	3	5	MFSJ7	8	4	7
	SFJS8	3	3	4	MFSJ8	9	4	8
	SFJS9	3	3	3	MFSJ9	11	4	8
	SFJS10	4	3	5	MFSJ10	12	4	8

Orta ve Büyük Boyutlu

Her bir test problemi için algoritma 10 defa koşturulmuş ve en iyi ulaşılan sonuç kaydedilmiş ve Çizelge 3'te karşılaştırmalı bir şekilde sunulmuştur.

Çizelge 3. Karşılaştırmalı sonuçlar

Problem	Bagheri ve ark., (2010)	Fattahi ve ark., (2007)	Fattahi ve ark., (2007)-2	Zandieh ve ark., (2008)	Önerilen TAA
SFJS1	66	66	66	66	66
SFJS2	107	107	107	107	107
SFJS3	221	221	221	221	221
SFJS4	355	355	355	355	355
SFJS5	119	119	119	119	119
SFJS6	320	320	320	320	320
SFJS7	397	397	397	397	397
SFJS8	253	253	253	253	253
SFJS9	210	210	215	210	210
SFJS10	516	516	516	516	516
MFJS1	468	469	488	468	468
MFJS2	448	468	457	448	457
MFJS3	468	533	599	466	482
MFJS4	554	618	703	554	583
MFJS5	527	625	674	514	559
MFJS6	625	717	816	634	704
MFJS7	879	1020	1048	881	1008
MFJS8	884	1030	1220	891	1026
MFJS9	1088	1105	1124	1094	1225
MFJS10	1267	1384	1546	1286	1431

Çizelge 3'ten görüleceği üzere önerilen algoritma küçük ve orta ölçekli problemler için optimum sonuç vermiştir. Büyük boyutlu problemler için kabul edilebilir çözümlere ulaşılmıştır.

SONUÇ

Bu çalışmada esnek atölye tipi çizelgeleme problemi için tekrarlı arama algoritması geliştirilmiştir. Toplam tamamlanma zamanı (Cmax) minimize edilmeye çalışılmıştır. Ele alınan problem, gezgin satıcı veya tek makine çizelgeleme problemi gibi permutasyon gösterimine uygun olmadığından probleme özgün yıkım-yeniden yapım operatörü geliştirilmiştir. Cmax değerini belirleyen operasyon kümesinden

(kritik operasyonlar) sökülen belirli sayıda (d_a) operasyon, kritik olmayan operasyonlar ile yer değiştirilmiştir. Başka bir deyişle kritik operasyonların atandığı makineler, kritik olmayanların atandığı makineler ile yer değiştirilmiştir. Ayrıca bu çalışmada özgün bir kabul kriteri tasarlanmıştır. İterasyona bağlı bu yaklaşımda ilk iterasyonda, en az ulaşılan en iyi sonuç seviyesinde ve daha iyi çözümler kabul edilirken, sonraki her iterasyonda bu kalite belirli oranda (q) düşürülmektedir. Küçük boyutlu problemler için optimum çözüme ulaşılmıştır. Ancak orta ve büyük boyutlu problemler için kabul edilebilir çözümlere ulaşılmıştır. Kritik operasyonlar üzerine kurulu bir şekilde geliştirilen yıkım-yeniden ile algoritmanın global arama kabiliyetinin geliştirildiği düşünülmektedir. Gelecek çalışma olarak çeşitli sıralama kuralları ile algoritmanın yerel arama kabiliyetinin artırılması planlanmaktadır.

KAYNAKLAR

- Al Aqel G, Li X, Gao L, Gong W, Wang R, Ren T, Wu G, 2018. Using Iterated Greedy with a New Population Approach for the Flexible Jobshop Scheduling Problem. In 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 1235-1239.
- Al Aqel G, Li X, Gao L, 2019. A modified iterated greedy algorithm for flexible job shop scheduling problem. Chinese Journal of Mechanical Engineering, 32(1): 21.
- Antczak A, Antczak P, Witkowski T, 2009. Using of evolving cellular automata for flexible job shop with makespan criterion. In 2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 297-302.
- Amjad MK, Butt SI, Kousar R, Ahmad R, Agha MH, Faping Z, ... Asgher U, 2018. Recent research trends in genetic algorithm based flexible job shop scheduling problems. Mathematical Problems in Engineering.
- Bagheri A, Zandieh M, Mahdavi I, Yazdani M, 2010. An artificial immune algorithm for the flexible job-shop scheduling problem. Future Generation Computer Systems, 26(4), 533-541.
- Bruker P, Schlie R, 1990 Job-shop scheduling with multi-purpose machines. Computing 45(4):369-375.
- Chaudhry I A, Khan A A, 2016. A research survey: review of flexible job shop scheduling techniques. International Transactions in Operational Research, 23(3): 551-591.
- Demir Y, İşleyen SK, 2014. An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations. International Journal of Production Research, 52(13): 3905-3921.
- Fattahi P, Saidi Mehrabad M, Jolai F, 2007. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. Journal of Intelligent Manufacturing, 18(3): 331-342.
- Gao J, Sun L, Gen M, 2008. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 35(9): 2892-2907.
- Gao K, Cao Z, Zhang L, Chen Z, Han Y, Pan Q, 2019. A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems. IEEE/CAA Journal of Automatica Sinica, 6(4): 904-916.
- Garey M R, Johnson D S, 1979. Computers and intractability (Vol. 174). San Francisco: freeman.
- Gen M, Cheng R, 1997. Genetic algorithms & engineering design. NewYork:Wiley.
- Jacobs L W, Brusco M J, 1995. A local search heuristic for large set-covering problems. Naval Research Logistics Quarterly, 42(7): 1129-1140
- Kacem I., Hammadi S, Borne P, 2002. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 32(1): 1-13.

- Lin S. W, Ying K C, Huang C Y, 2013. Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm. *International Journal of Production Research*, 51(16): 5029-5038.
- Pezzella F, Morganti G, Ciaschetti G, 2008. A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research*, 35(10): 3202-3212.
- Pinedo M, 2002. *Scheduling theory, algorithms, and systems*. Englewood Cliffs, NJ: Prentice-Hall.
- Riahi V, Chiong R, Zhang Y, 2020. A new iterated greedy algorithm for no-idle permutation flowshop scheduling with the total tardiness criterion. *Computers & Operations Research*, 117, 104839.
- Rodriguez FJ, Lozano M, Blum C, GarcíA-MartíNez C, 2013. An iterated greedy algorithm for the large-scale unrelated parallel machines scheduling problem. *Computers & Operations Research*, 40(7): 1829-1841.
- Ruiz R, Stutzle T, 2007. A Simple and Effective Iterated Greedy Algorithm for the Permutation Flowshop Scheduling Problem, *European Journal of Operational Research*, 177, 2033–2049.
- Xie J, Gao L, Peng K, Li X, Li H, 2019. Review on flexible job shop scheduling. *IET Collaborative Intelligent Manufacturing*, 1(3): 67-77.
- Zandieh M, Mahdavi I, Bagheri A, 2008. Solving the flexible job-shop scheduling problem by a genetic algorithm. *JApSc*, 8(24): 4650-4655.
- Zhang G, Gao L, Shi Y, 2011. An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Systems with Applications*, 38(4): 3563-3573.