Pamukkale Univ Muh Bilim Derg, 24(1), 130-140, 2018

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Basit düz ve U-tipi montaj hattı dengeleme problemleri için diferansiyel evrim algoritması

A differential evolution algorithm for simple straight and U-type assembly line balancing problems

Feriştah ÖZÇELİK[1*]

[1]Endüstri Mühendisliği Bölümü, Mühendislik-Mimarlık Fakültesi, Eskişehir Osmangazi Üniversitesi, Eskişehir, Türkiye.
fdurmaz@ogu.edu.tr

Geliş Tarihi/Received: 28.11.2016, Kabul Tarihi/Accepted: 12.04.2017
doi: 10.5505/pajes.2017.47487
* Yazışılan yazar/Corresponding author
Araştırma Makalesi/Research Article

Öz

Montaj hattı, seri olarak birbirine bağlı istasyonlardan oluşan bir akış tipi üretim sistemidir. Montaj hatlarının etkin olarak tasarımı, standart ürünlerin büyük miktarlarda üretiminde oldukça önemlidir. Bu çalışmada, düz ve U-tipi basit montaj hattı dengeleme problemlerinin çözümü için bir diferansiyel evrim algoritması geliştirilmiştir. Popülasyon temelli evrimsel bir algoritma olan diferansiyel evrim algoritması, son yıllarda eniyileme problemlerinin çözümünde etkin olarak kullanılan bir yöntem olarak karşımıza çıkmaktadır. Önerilen algoritmanın çözüm başarısı, literatürde yaygın olarak kullanılan çok sayıda test problemi kullanılarak gerçekleştirilen deneyler ile değerlendirilmiştir. Sonuçlar algoritmanın etkinliğini göstermektedir.

Anahtar kelimeler: Basit montaj hattı dengeleme, U-tipi hatlar, Diferansiyel evrim algoritması, Evrimsel algoritma

Abstract

An assembly line is a flow-oriented production system in which the productive units performing the operations, referred to as stations, are aligned in a serial manner. Design of efficient assembly lines has considerable importance for the production of high-quantity standardized products. In this paper, a differential evolution algorithm is proposed to solve simple straight and U-type assembly line balancing problems. As a population-based evolutionary algorithm, differential evolution algorithm is seen as an effective method to solve optimization problems in recent years. A computational study is conducted by solving a large number of benchmark problems available in the literature to compare the performance of the proposed approach. The results show that the proposed approach performs quite effectively.

Keywords: Simple assembly line balancing, U lines, Differential evolution algorithm, Evolutionary algorithm

1 Giriş

Malzemelerin, akış hattı boyunca işgücü veya donanımdan yararlanılarak iletildiği ve parça üzerindeki işlemlerin/görevlerin; aralarındaki -öncelik ilişkileri ve çevrim süresi gibi - kısıtlar göz önüne alınarak birleştirilmesiyle oluşturdukları istasyonların, yine bir hat boyunca sıralanmasıyla oluşan sisteme, 'montaj hattı' adı verilmektedir [1]. Montaj hatlarının tasarlanmasında ve talep değişikliklerine göre üretim hızının tekrar ayarlanmasında ortaya çıkan en önemli problem, montaj hattı dengeleme problemidir. Montaj hattı dengeleme problemi, bazı kısıtlar altında bir veya birden fazla amaç gözetilerek görevlerin istasyonlara atanması problemidir. Genel olarak montaj hattı dengeleme problemine ilişkin temel kısıtlar şöyle sıralanabilir:

(i) Bir görev sadece bir istasyona atanmalıdır (atama kısıtı),

(ii) Bir görevin bir istasyona atanabilmesi için o görevin bütün öncelik ilişkilerinin sağlanması gerekir (öncelik ilişkileri kısıtı),

(iii) Herhangi bir istasyonda yer alan görevlerin süreleri toplamı çevrim süresini aşamaz (çevrim süresi kısıtı).

Montaj hattı dengeleme probleminin en basit hali, basit montaj hattı dengeleme (BMHD) problemi olarak adlandırılan tek modelli ve deterministik görev süreli montaj hattı dengeleme problemidir. Tüm parametrelerin kesin olarak bilindiği BMHD probleminin, çevrim süresi verildiğinde iş istasyonu sayısının enküçüklenmesinin amaçlandığı tip-1 ve istasyon sayısı verildiğinde çevrim süresinin enküçüklenmesinin amaçlandığı tip-2 olmak üzere iki türü vardır.

Seri üretimin temelini oluşturan geleneksel düz montaj hatları müşteri beklentileri, rekabet faktörleri ve teknolojik ilerlemelere bağlı olarak tasarım açısından değişik şekillerde kullanılmıştır. Bunlardan birisi olan U-tipi montaj hattı, Toyota'da Tam Zamanında Üretim Sisteminin uygulanması için gereklerden biri olarak ortaya çıkmıştır. Düz montaj hatları ile U-tipi montaj hatlarının arasındaki temel farklılık, montaj hattının yerleşimidir. Düz montaj hatlarında istasyonlar düz bir çizgi şeklinde yerleştirilirken, U-tipi montaj hatları hattın giriş ve çıkış yönleri aynı tarafta olacak biçimde U (⊃) şeklinde tasarlanmıştır. U-tipi montaj hatlarındaki esneklik ile değişen üretim koşullarına uyum sağlanması kolaylaşmaktadır. Bu hatlarda, operatör sayısında artış veya azalış yapılarak üretim hızı istenen seviyede tutulabilmektedir. Ayrıca, operatörler aynı anda birçok iş istasyonunu görebildiği için aralarındaki iletişim ve işbirliği artar. Hatta problemler ortaya çıktığında operatörler hızlı bir şekilde hareket edip yardımlaşarak sorunu çözebilirler [2]. U-tipi hatlarla düz hatlara kıyasla daha az sayıda istasyon ile montaj hattını dengelemek mümkündür.

Hatların fiziksel farklılığı nedeniyle, geleneksel hatlarda uygulanan yöntem ve tanımlar, U-tipi hatlar için farklılık gösterebilir. Geleneksel hat dengeleme problemi ile U-tipi hattı dengeleme problemi arasındaki anahtar fark; geleneksel hat dengeleme probleminde atanabilir görevler kümesinin öncülleri atanmış görevlerden oluşması ve görevlerin bu kümeden seçilerek ilgili istasyona atanmasıdır. U-tipi hat

dengeleme probleminde ise atanabilir görevler kümesi öncülleri atanmış görevler kümesi ile ardılları atanmış görevler kümesinin birleşiminden oluşur. İstasyona atanacak görevler bu kümeden seçilir.

Basit düz montaj hattı dengeleme (BDMHD) problemleri gibi, basit U-tipi montaj hattı dengeleme (BUMHD) problemleri de NP-zor bir yapıya sahiptir [3]. BDMHD problemi ilk kez Salveson [4] tarafından modellenmiş ve günümüze kadar pek çok araştırmacı tarafından yoğun olarak çalışılmıştır. Bu çalışmalarda kullanılan çözüm yöntemleri, kesin çözüm yöntemleri [4]-[13], sezgisel algoritma [14]-[19], genetik algoritmalar [20]-[25], yasaklı arama [26]-[28], karınca kolonisi sistemleri [29]-[30], genetik programlama [31] ve tavlama benzetimidir [32]-[36]. BUMHD problemi ise ilk kez Miltenburg and Wijngaard [37] tarafından modellenmiş, problemin ilk tamsayılı modelini de Urban [38] geliştirmiştir. BUMHD konusunda düz hatlara kıyasla daha az sayıda çalışma bulunmaktadır [3],[35],[36],[39]-[44].

Diferansiyel evrim algoritması (DEA) son yıllarda eniyileme problemlerinin çözümünde etkin olarak kullanılan bir yöntem olarak karşımıza çıkmaktadır. DEA, Storn ve Price [45] tarafından 1995 yılında geliştirilen ve özellikle sürekli eniyileme problemlerinin çözümünde kullanılan popülasyon temelli sezgisel bir algoritmadır. Montaj hattı dengeleme problemlerinin çözümünde DEA kullanımı az sayıda çalışmada ele alınmasına rağmen sonuçlar DEA'nın bu problemlere başarıyla uygulandığını göstermektedir. Nearchou [46], tip-2 basit montaj hattı dengeleme probleminin çevrim süresinin enküçüklenmesi amacıyla çözümü için ve Mozdgir ve diğ. [47] düzgünlük indeksinin enküçüklenmesi amacıyla çözümü için DEA geliştirmiştir. Nearchou [48] ve Nourmohammadi ve Zandieh [49] ise çok amaçlı BDMHD-2 probleminin çözümü için DEA geliştirmişlerdir. Nearchou [48] asıl amaç olarak çevrim süresinin enküçüklenmesini, ikincil amaçlar olarak da dengeleme gecikmesinin ve düzgünlük indeksinin enküçüklenmesini dikkate almıştır. Nourmohammadi ve Zandieh [49] çevrim süresi ve düzgünlük indeksinin enküçüklenmesi amaçlarıyla probleme çözüm aramıştır. Nearchou [48] çok amaçlı problemi, amaçları ağırlıklandırma yöntemi ile birleştirip tek amaçlı yapıya dönüştürerek çözmüştür. Nourmohammadi ve Zandieh [49] ise bir sonraki popülasyonda yer alacak bireyleri seçerken amaçları ayrı ayrı ele alma imkânı sağlayan Pareto baskınlık kavramına dayanan bir kabul planı ve TOPSIS yöntemine dayalı bir değerlendirme mekanizması kullanmıştır. DEA sürekli problemler için geliştirilen bir yöntem olduğundan hat dengeleme problemi gibi kesikli bir problemin çözümünde kullanılırken kromozomlar direkt olarak çözümleri göstermediği için kromozomları çözüme dönüştürecek bir mekanizmaya ihtiyaç vardır. Bu çözüm mekanizması algoritmanın başarısını etkilediği için doğru mekanizma kullanımı oldukça önemlidir. 2016 yılında Zhang ve diğ. [50] tip-2 basit montaj hattı dengeleme probleminin çözümü için tamsayı kodlamalı diferansiyel evrim algoritması ismini verdikleri bir algoritma geliştirmişlerdir. Bu algoritmanın özelliği kesikli yapıda olması yani kromozomların tamsayılı değerler alması nedeniyle herhangi bir dönüşüme ihtiyaç duymamasıdır. Yazarlar ayrıca yeni mutasyon ve çaprazlama operatörleri de önermişlerdir.

BDMHD-1 probleminin çözümünde diferansiyel evrim algoritması ilk kez Nearchou [51] tarafından kullanılmıştır. Nearchou [51] kromozomları çözüme dönüştürürken öncelikle görevlerin istasyonlara atanma sırasını belirlemekte ve bu sıralamaya göre atamaları yapmaktadır. Sıralama, alt aralık (sub-range) yöntemine göre belirlenmektedir. Bu yöntemde [0-1] aralığı, görev sayısı kadar eşit aralığa bölünmektedir. Daha sonra kromozom değerleri karşı geldikleri aralığa göre görev numaralarına dönüşmektedir. Ancak sıralama belirlenirken öncelik kısıtları dikkate alınmadığı için olurlu olmayan çözümler türeyebilmektedir. Yazar, olurlu olmayan çözümler için bir tamir mekanizması kullanmaktadır. Pitakaso [52] BDMHD-1 problemi için önerdiği DEA'nda çözümleri oluştururken öncelik kısıtlarını dikkate almaktadır. Öncelik kısıtını sağlayan birden fazla görev var ise istasyona hangi görevin atanacağı rulet çemberi seçim yöntemine göre belirlenmektedir. Görevler seçilirken kromozomda aldıkları değerler kullanılmakta, değeri büyük olana daha çok seçilme şansı verilmektedir. Bu yöntemde bir kromozom her zaman aynı çözümü değil farklı çözümleri de (istasyon atamalarına) temsil edebilmektedir. Yazarlar, farklı mutasyon ve çaprazlama yöntemlerinin performansını da test etmiştir. Pitakaso ve Sethanan [53] her bir görevin önceden bilinen belirli bir makinede gerçekleştirileceği ve alan yetersizliği ya da işçi becerisi gibi nedenlerle her istasyona atanabilecek makine sayısına bir üst sınırın olduğu BDMHD-1 problemini ele almış ve problemin çözümü için bir DEA geliştirmiştir.

Diferansiyel evrim algoritmasının montaj hattı dengeleme problemlerinin çözümündeki uygulamalara bakıldığında çalışmaların tamamının düz hatlar ve çoğunluğunun tip-2 basit montaj hattı dengeleme problemi için olduğu görülmektedir. Bu çalışmada, tip-1 düz ve U basit montaj hattı dengeleme problemlerinin çözümü için bir diferansiyel evrim algoritması geliştirilmiş ve önerilen algoritmanın çözüm başarısı literatürde yaygın olarak kullanılan çok sayıda test problemi kullanılarak gerçekleştirilen deneyler ile değerlendirilmiştir. Test problemlerine dayanarak yapılan karşılaştırmalar, bu çalışmayla önerilen algoritmanın literatürdeki algoritmalardan daha başarılı olduğunu göstermiştir. Önerilen algoritmanın temeldeki üstünlüğü de, sağlam kromozom yapısından kaynaklanmaktadır. Gerçekten de, kullanılan kromozom yapısı hem öncelik ilişkilerine ağırlık verebilmekte, hem tamir gerektirmemekte hem de dönüşümde rassallığa yer vermediği için kendisi ile atama şekli arasında deterministik bir haritalama sağlamaktadır.

Çalışmanın izleyen bölümünde geliştirilen diferansiyel gelişim algoritması açıklanmış, üçüncü bölümde deneysel sonuçlar sunulmuş ve son bölümde ise elde edilen sonuçlar ve gelecek çalışmalara yönelik öneriler verilmiştir.

2 Diferansiyel evrim algoritması

DEA, diğer evrimsel algoritmalar gibi rassal olarak türetilmiş bir başlangıç popülasyonu ile başlar. Her nesilde daha kaliteli bireylere sahip yeni popülasyonlar elde etmek amacıyla popülasyona mutasyon, çaprazlama ve seçim operatörleri uygulanmaktadır. Bu evrimsel süreç, önceden belirlenmiş bir durma koşulu sağlanıncaya kadar devam etmektedir.

2.1 Başlangıç popülasyonunun türetilmesi

DEA rassal olarak türetilen N_p adet D-boyutlu gerçek değerli vektörden oluşan bir popülasyon ile başlar. Kromozom olarak da isimlendirilebilen her vektör, ilgilenilen probleme aday bir çözümü temsil eder. G neslindeki popülasyonun i. bireyi X_{iG} aşağıdaki şekilde gösterilebilir:

$$X_{iG} = [x_{1iG}, x_{2iG}, \dots, x_{DiG},], i = 1,2, \dots, N_p \quad (1)$$

2.2 Mutasyon

En yaygın mutasyon operatöründe her bir hedef vektör X_{iG} için mutant vektör V_{iG} aşağıdaki şekilde türetilmektedir:

$$V_{iG} = X_{aG} + F.(X_{bG} - X_{cG}) \quad (2)$$

a, b ve c indisleri $[1, N_p]$ aralığından rassal olarak seçilmiş birbirinden ve i indisinden farklı tamsayılardır. Bu indisler, her bir mutant vektör için rassal olarak ayrı ayrı türetilmektedir. Ölçek faktörü F, fark vektörlerinin ölçeklendirilmesinde kullanılan [0,2] aralığında değer alan bir kontrol parametresidir.

2.3 Çaprazlama

Aday bireyin (U_{iG}) bileşenleri aşağıdaki ilişkiye göre ya hedef vektörden (X_{iG}) ya da mutant vektörden (V_{iG}) alınmaktadır.

$$U_{jiG} = V_{jiG}; eğer\ rassal \leq CR\ veya\ j = IntRnd(1, D) \quad (3)$$

$$U_{jiG} = X_{jiG}; diğer\ durumda \quad (4)$$

$Rassal$, [0,1] aralığında bir rassal sayı ve $IntRnd(1, D)$, $[1, D]$ aralığında rassal olarak türetilmiş bir tamsayıyı göstermektedir. (3) numaralı ifadedeki ikinci koşul, en az bir bileşenin mutant vektörden alınmasını garanti etmektedir. CR, çaprazlama oranı olarak isimlendirilen bir kontrol parametresidir.

2.4 Seçim

Seçim, yeni nesilin $(G + 1)$ bireylerinin türetilmesi sürecidir. Bir sonraki nesile hedef vektörün mü yoksa aday vektörün mü taşınacağına amaç fonksiyonu değerlerine göre karar verilmektedir. Enküçükleme problemlerinde eğer aday vektörün (U_{iG}) amaç fonksiyonu değeri hedef vektörünkine (X_{iG}) eşit veya daha az ise sonraki nesilde hedef vektör aday vektörle değiştirilir. Aksi halde, hedef vektör sonraki nesilde de yaşamaya devam eder. Böylece popülasyon ya daha iyiye gider ya da aynı kalır, fakat hiçbir zaman kötüleşmez. Bu aç gözlü mekanizma aşağıdaki şekilde ifade edilebilir.

$$X_{i.G+1} = U_{iG}; eğer\ f(U_{iG}) \leq f(X_{iG}) \quad (5)$$

$$X_{i.G+1} = X_{iG}; diğer\ durumda \quad (6)$$

3 Geliştirilen diferansiyel evrim algoritması

Bu bölümde basit düz ve U-tipi montaj hattı dengeleme problemlerinin çözümü için önerilen diferansiyel evrim algoritması anlatılmıştır.

3.1 Çözümlerin gösterimi

Önerilen DEA'nda çözümler, her biri [0,1] aralığında gerçek değer alan D (görev sayısı) boyutlu vektörler ile gösterilmiştir (Şekil 1). Vektörün her bir elemanı görev önceliğini göstermektedir. Örneğin iş istasyonlarına atanacak 7 adet görev varsa vektör boyutu 7 (vektörün 7 elemanı var) olacaktır. Şekil 1'den görülebileceği gibi görev 1'nin önceliği 0.01 ve görev 2'nin önceliği 0.03'tür. Başlangıç popülasyonu oluşturmak için N_p adet vektör rassal olarak türetilmiştir. Popülasyon büyüklüğü N_p, görev sayısına eşit olarak alınmıştır.

Görev indeksi	**1**	**2**	**3**	**4**	**5**	**6**	7
Görev önceliği	0.01	0.03	0.86	0.20	0.27	0.67	0.32

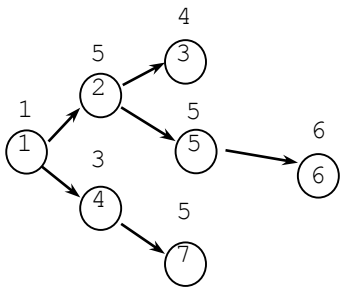
Şekil 1: Çözüm vektörünün yapısı.

3.2 Görevlerin atanması

Çözüm vektörüne göre görevlerin istasyonlara atanması (kromozomların çözüme dönüştürülmesi), aşağıdaki yordama göre yapılmıştır.

Yordam Görev Atama
başla
  mevcut istasyondaki boş süreyi ve görevler arasındaki öncelik ilişkilerini dikkate alarak atanabilir görevler kümesini belirle,
  $k = 0$;
tüm görevler atanıncaya kadar tekrar et
  k. istasyonu aç, $k = k + 1$;
  atanabilir görevler kümesi boş küme oluncaya kadar tekrar et
    atanabilir görevler kümesindeki görevlerden enbüyük önceliğe sahip görevi seç
    seçilen görevi k. istasyona ata
    k. istasyonun boş süresini ve atanabilir görevler kümesini güncelle
  dur
dur
eldeki çözüm için amaç foksiyonunun değerini hesapla
dur

Atanabilir görevler kümesinin elemanları, herhangi bir istasyona atanmamış görevlerden görev süresi, istasyonun boş süresinden küçük veya eşit olan ve öncelik kısıtlarını sağlayanlardan oluşmaktadır. Öncelik kısıtlarını sağlayan görevler ise düz hatlar için öncülleri atanmış görevler ve U-tipi hatlar için öncülü ya da ardılı atanmış görevlerdir.



Şekil 2: Örnek problem için öncelik diyagramı.

Görev atama yordamının nasıl çalıştığını, 7 görevli Mertens problemi üzerinde inceleyelim. Şekil 2'de düğümler içerisindeki rakamlar görevleri, düğümlerin üstündeki rakamlar ise görev sürelerini göstermektedir. İki düğüm arasındaki yönlü ok, öncelik ilişkisini ifade etmektedir. Çevrim süresi 10 olarak alınsın. Şekil 1'deki kromozom için görev atama yordamının çalışma adımları Tablo 1'de verilmiştir. Başlangıçta herhangi bir atama yapılmadığı için 1. istasyonda çevrim süresi kadar boş süre vardır. Öncelik kısıtı açısından istasyona atanabilecek tek görev vardır: görev 1. Görev 1'in süresi, istasyonun boş süresinden küçük olduğundan atanabilir görevler kümesinin tek elemanı görev 1'dir. 1. istasyona görev 1'in atanması sonucunda ilgili istasyonda kalan boş süre 9'dur. Algoritma tüm görevler atanıncaya kadar bu şekilde tekrar etmektedir. Tablo 1'den görüleceği üzere Şekil 1'deki kromozom, 1, 4 ve 7 numaralı görevlerin 1. istasyona, 2 ve 3 numaralı görevlerin 2. istasyona, 5 numaralı görevin 3. istasyona ve 6 numaralı görevin 4. istasyona atandığı dört istasyonlu bir çözümü temsil etmektedir.

Tablo 1: Görev atama yordamının adımları.

Tekrar	İst.	Boş süre	Atanabilir görevler kümesi	Görev öncelikleri	Atanan görev
1	1	10	{1}	{0.01}	1
2	1	9	{2,4}	{0.03,0.20}	4
3	1	6	{2,7}	{0.03,0.32}	7
4	2	10	{2}	{0.03}	2
5	2	5	{3,5}	{0.86,0.27}	3
6	3	10	{5}	{0.27}	5
7	4	10	{6}	{0.67}	6

3.3 Amaç fonksiyonu

Bu çalışmada ele alınan tip-1 basit düz ve U-tipi montaj hattı dengeleme problemlerinin amacı, istasyon sayısının enküçüklenmesidir. Ancak aynı istasyon sayısına sahip iki farklı çözüm incelendiğinde bunlardan birisinin diğerinden daha dengeli olduğu görülebilir. Örneğin dört istasyonlu iki çözümden istasyon süreleri 40-40-30-30 olan 20-50-30-40 olandan daha dengelidir [35]. Bu nedenle literatürde ilgilenilen problem için düzgünlük indeksi (DI) [31],[35],[36] ve işyükü değişimi (V) [25, 31, 36] gibi farklı performans ölçütlerinin kullanıldığı görülmüştür. DI veya V'nin enküçüklenmesi ile istasyonlar arasındaki iş yükü farklılığının azaltılması ve bu iş yükünün istasyonlara mümkün olduğunca eşit dağıtılması amaçlanmaktadır. m istasyon sayısını, C çevrim süresini, ST_{enb} en büyük istasyon süresini ve ST_s s istasyonunun süresini gösterirken bu performans ölçütleri aşağıdaki şekilde hesaplanmaktadır:

$$f(DI) = \sqrt{\frac{\sum_{s=1}^{m}(ST_{enb} - ST_s)^2}{m}} \quad (7)$$

$$f(V) = \sqrt{\frac{\sum_{s=1}^{m}(\frac{ST_s}{ST_{enb}} - \frac{\sum_{s=1}^{m} ST_s}{m})^2}{m}} \quad (8)$$

Bu çalışmada, önerilen DEA'nın amaç fonksiyonu olarak iş yükünü dengeli dağıtmayı hedefleyen *enk V* ve *enk DI* olmak üzere iki farklı amaç fonksiyonu ayrı ayrı kullanılmıştır.

3.4 Geliştirilen algoritmanın genel yapısı

Geliştirilen algoritmanın yapısı izleyen şekildedir:

başla
parametreleri gir;
başlangıç popülasyonunu oluştur;
başlangıç popülasyonunu değerlendir;
çözümler yakınsayıncaya kadar tekrar et
başla
mutasyon;
çaprazlama;
seçim;
dur
dur

Algoritma popülasyon büyüklüğü N_p, ölçek faktörü F ve çaprazlama oranı CR gibi diferansiyel evrim algoritması parametrelerini okuyarak başlamaktadır. Daha sonra popülasyon büyüklüğü kadar birey rassal olarak türetilmekte ve amaç fonksiyonu değerleri belirlenmektedir. Algoritma yakınsayıncaya (belirli sayıda nesilde çözümlerde iyileşme olmayıncaya) kadar popülasyona mutasyon, çaprazlama ve seçim operatörleri uygulanmaktadır.

Geliştirilen algoritmanın adımları Bölüm 3.2'de verilen örnek problem kullanılarak açıklanmış ve Şekil 3'te gösterilmiştir. Mevcut popülasyona öncelikle mutasyon operatörü uygulanmaktadır. Hedef vektör X_{1G} olsun. Hedef vektör için rassal olarak belirlenen a, b ve c indisleri sırasıyla 5, 4, 2 ve ölçek faktörü F'in değeri 0.9 iken mutant vektör $V_{1G} = X_{5G} + 0.9\,(X_{4G} - X_{2G})$ eşitliği ile türetilmektedir. Çaprazlama oranı $CR = 0.4$, $j = 3$ ve türetilen rassal sayılar $(0.32,0.67,0.78,0.56,0,18,0.45,0.03)$ iken aday vektörün birinci, üçüncü, beşinci ve yedinci elemanları mutant vektörden, diğerleri ise hedef vektörden gelmektedir. Aday vektörün amaç fonksiyonu değeri (DI) 0.577 hedef vektörün amaç fonksiyonu değeri 2.5'dan daha iyi olduğu için bir sonraki neslin ilk bireyi aday vektör olacaktır. Bu işlemler, durma koşulu sağlanıncaya kadar tekrarlanacaktır.

4 Deneysel sonuçlar

Geliştirilen DEA, Delphi Pascal dilinde kodlanmış ve test problemleri 2.67 GHz CPU, 3 GB RAM özelliklerine sahip Intel Core 2 Quad PC kullanılarak test edilmiştir. Test problemleri http://www.assembly-line-balancing.de adresinden alınmıştır. Algoritmanın bütün parametreleri literatürdeki öneriler doğrultusunda deneysel olarak belirlenmiş ve tüm test problemleri için sabitlenmiştir. Popülasyon büyüklüğünün değeri görev sayısına eşit ($N_p = D$), ölçek faktörü F'in değeri Rönkkönen [54] tarafından önerildiği gibi 0.9 ve çaprazlama oranı Storn ve Price [45] tarafından önerildiği gibi 0.1 olarak alınmıştır. Durma koşulu olarak da *son 2000 nesilde iyileşme olmaz ise durdur* koşulu kullanılmıştır. Baykasoğlu ve Özbakır [31], Özcan ve Toklu [36] ve Pitakaso [52] her bir test problemini önerdikleri algoritma ile 5 kez çözmüş ve elde edilen eniyi sonuçları vermişlerdir. Bu çalışmada da benzer bir yöntem izlenmiş ve karşılaştırmalar eniyi değerler üzerinden yapılmıştır. Çözüm süreleri, test problemlerine göre 0 ile 580 saniye arasında değişmektedir.

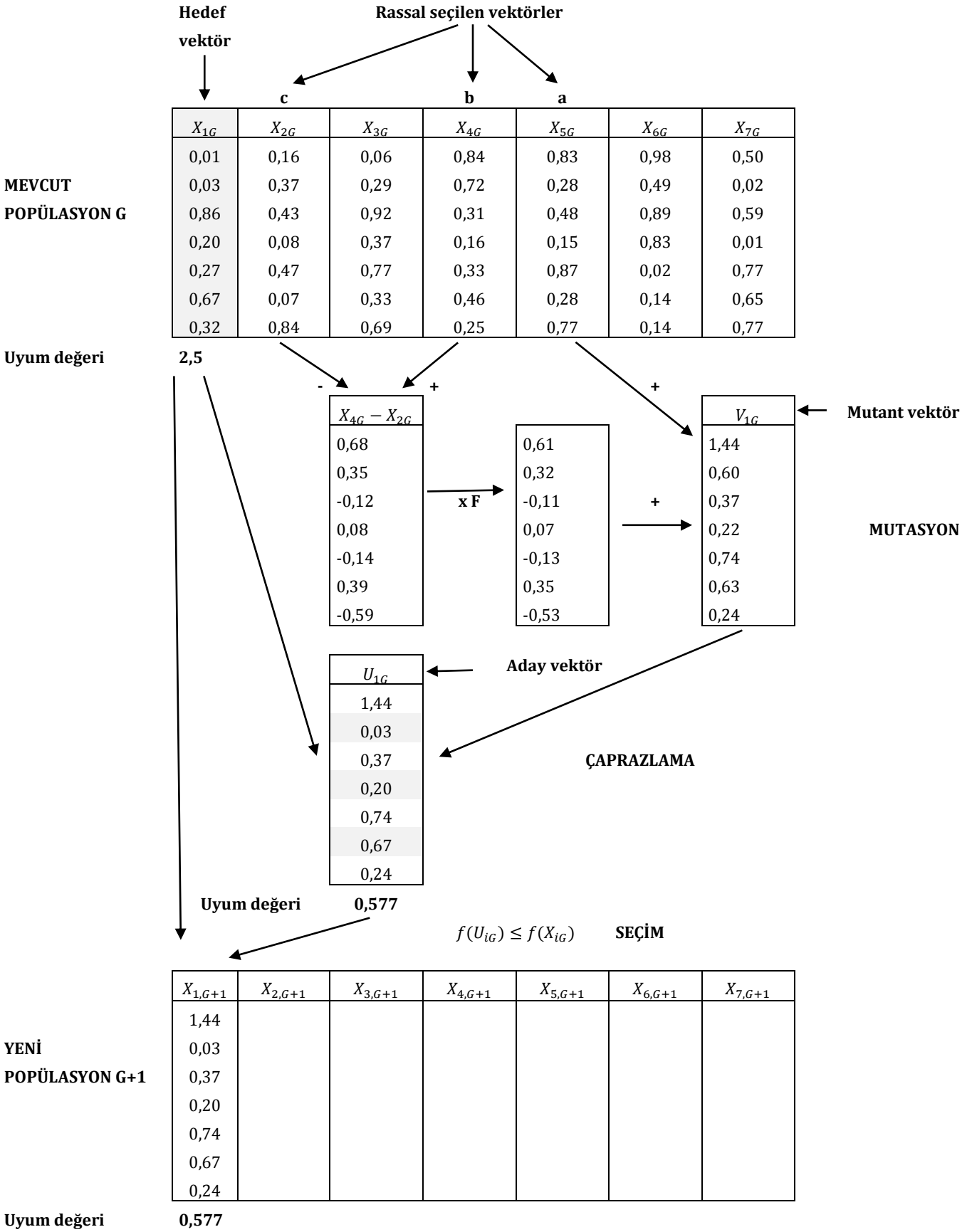
Geliştirilen algoritmanın performansı ilgilenilen problemin çözümü için literatürde önerilmiş diğer diferansiyel evrim algoritması ve genetik algoritma, tavlama benzetimi gibi diğer yöntemlerle karşılaştırılarak test edilmiştir.

4.1 Diğer DEA ile karşılaştırma

Nearchou [51] algoritmasını Talbot'un test problemini kullanarak test etmiştir. Aynı problemler Pitakaso [52] tarafından da kullanılmıştır. Bu problem grubu, görev sayısı 7 ile 111 arasında ve çevrim süresi 6 ile 17067 arasında değişen 64 problemden oluşmaktadır. Karşılaştırma sonuçları Tablo 2'de verilmiştir. Tablonun ilk satırında çözülen 64 test probleminden kaç tanesinde, ikinci satırında da yüzde kaçında eniyi çözüme ulaşıldığı bilgisi yer almaktadır. Üçüncü satırda ise eniyi çözümden sapma yüzdesi verilmiştir. Tablodan da görülebileceği gibi Nearchou [51] problemlerin %95.31'inde eniyi sonuçlara ulaşabilmişken Pitakaso [52] ve önerilen DEA %100'ünde ulaşmıştır. Testlerde kullanılan bilgisayarlar farklı özelliklerde olduğundan ve Pitakaso [52] testlerin yapıldığı bilgisayar hakkında herhangi bir bilgi vermediğinden süre açısından bir karşılaştırma yapılamamıştır.

Tablo 2: Diğer DEA ile karşılaştırma sonuçları.

	Nearchou [51]	Pitakaso [52]	DEA
# eniyi	61	64	64
% eniyi	%95.31	%100	%100
% sapma	%4.67	%0	%0



Şekil 3: DEA'nın adımları.

4.2 Diğer yöntemler ile karşılaştırma

Önerilen algoritmanın performansını değerlendirmek için çok amaçlı genetik algoritma (MOGA [25]), genetik programlama (GP [31]), çok amaçlı tavlama benzetimi algoritması (MRMOSA [35]) ve çok amaçlı melez iyileştirme sezgiselinin (MOHIH [36]) sonuçları kullanılmıştır. Sonuçlar Tablo 3 ila Tablo 8'de verilmiştir.

Tablo 3: BDMHD problemlerinin *enk DI* amaç fonksiyonuyla çözümü sonucunda elde edilen sonuçlar.

Problem	Görev sayısı	C	Opt m	MRMOSA		MOHIH		GP		DEA				
				m	DI	m	DI	m	DI	m	eniyi	ort	enkötü	std sapma
Merten	7	6	6	6	**1.354**	6	**1.354**	6	**1.354**	6	**1.354**	1.354	1.354	0.000
		7	5	5	**1.414**	5	**1.414**	5	**1.414**	5	**1.414**	1.414	1.414	0.000
		8	5	5	**1.414**	5	**1.414**	5	**1.414**	5	**1.414**	1.414	1.414	0.000
		10	3	3	**0.577**	3	**0.577**	3	**0.577**	3	**0.577**	0.577	0.577	0.000
		15	2	2	**0.707**	2	**0.707**	2	**0.707**	2	**0.707**	0.707	0.707	0.000
		18	2	2	**3.535**	2	**3.535**	2	**3.535**	2	**3.535**	3.535	3.535	0.000
Jaeschke	9	6	8	8	**1.695**	8	**1.695**	8	**1.695**	8	**1.695**	1.695	1.695	0.000
		7	7	7	**2.000**	7	**2.000**	7	**2.000**	7	**2.000**	2.000	2.000	0.000
		8	6	6	**2.345**	6	**2.345**	6	**2.345**	6	**2.345**	2.345	2.345	0.000
		10	4	4	**0.866**	4	**0.866**	4	**0.866**	4	**0.866**	0.866	0.866	0.000
		18	3	3	**7.505**	3	**7.505**	3	**7.505**	3	**7.505**	7.505	7.505	0.000
Jackson	11	7	8	8	**1.658**	8	**1.658**	8	**1.658**	8	**1.658**	1.658	1.658	0.000
		9	6	6	**1.732**	6	**1.732**	6	**1.732**	6	**1.732**	1.732	1.732	0.000
		10	5	5	**1.095**	5	**1.095**	5	**1.095**	5	**1.095**	1.095	1.095	0.000
		13	4	4	**0.707**	4	**0.707**	4	**0.707**	4	**0.707**	0.707	0.707	0.000
		14	4	4	**0.707**	4	**0.707**	4	**0.707**	4	**0.707**	0.707	0.707	0.000
		21	3	3	**5.802**	3	**5.802**	3	**5.802**	3	**5.802**	5.802	5.802	0.000
Mitchell	21	14	8	8	**1.060**	8	**1.060**	8	**1.060**	8	**1.060**	1.060	1.060	0.000
		15	8	8	**2.318**	8	**2.318**	8	**2.318**	8	**2.318**	2.318	2.318	0.000
		21	5	5	**0.000**	5	**0.000**	5	**0.000**	5	**0.000**	0.000	0.000	0.000
Heskiaoff	28	138	8	8	9.013	8	5.831	8	5.701	8	**4.847**	5.274	5.701	0.404
		205	5	5	**0.447**	5	**0.447**	5	**0.447**	5	**0.447**	0.447	0.447	0.000
		216	5	5	4.098	5	2.490	5	2.489	5	**1.414**	1.414	1.414	0.000
		256	4	4	**0.000**	4	**0.000**	4	**0.000**	4	**0.000**	0.000	0.000	0.000
		324	4	4	73.880	4	**62.948**	4	**62.948**	4	**62.948**	62.948	62.948	0.000
		342	3	3	**0.816**	3	**0.816**	3	**0.816**	3	**0.816**	0.816	0.816	0.000
Sawyer	30	25	14	14	2.360	14	2.204	14	2.204	14	**2.171**	2.171	2.171	0.000
		27	13	13	1.709	13	**1.358**	13	**1.358**	13	**1.358**	1.358	1.358	0.000
		30	12	12	2.415	12	2.415	12	2.415	12	**2.345**	2.345	2.345	0.000
		36	10	10	2.049	10	**1.844**	10	1.897	10	**1.844**	1.876	1.897	0.029
		41	8	8	**0.707**	8	**0.707**	8	**0.707**	8	**0.707**	0.707	0.707	0.000
		54	7	7	2.653	7	2.390	7	2.390	7	**2.204**	2.255	2.268	0.029
		75	5	5	9.316	5	3.950	5	**3.741**	5	**3.741**	3.908	3.950	0.093
Kill.&Wes.	45	57	10	10	1.000	10	**0.894**	10	1.000	10	**0.894**	0.894	0.894	0.000
		79	7	7	**0.377**	7	**0.377**	7	**0.377**	7	**0.377**	0.377	0.377	0.000
		92	6	6	**0.000**	6	**0.000**	6	**0.000**	6	**0.000**	0.000	0.000	0.000
		110	6	6	25.046	6	18.876	6	**9.146**	6	**9.146**	9.146	9.146	0.000
		138	4	4	**0.000**	4	**0.000**	4	**0.000**	4	**0.000**	0.000	0.000	0.000
		184	3	3	**0.000**	3	**0.000**	3	**0.000**	3	**0.000**	0.000	0.000	0.000
Tonge	70	176	21	21	8.423	21	8.696	21	7.060	21	**3.915**	4.944	5.740	0.750
		364	10	10	4.795	10	5.745	10	4.381	10	**3.405**	3.554	3.606	0.087
		410	9	9	6.896	9	6.200	9	6.027	9	**3.636**	3.648	3.697	0.027
		468	8	8	15.049	8	9.300	8	11.169	8	**8.093**	8.130	8.185	0.050
		527	7	7	14.540	7	9.848	7	10.816	7	**5.555**	5.606	5.707	0.062
Arcus1	83	5048	16	16	284.790	16	260.252	16	254.874	16	**245.220**	246.216	248.938	1.542
		5853	14	14	293.290	14	225.637	14	161.635	14	**91.772**	123.150	146.306	26.965
		6842	12	12	522.810	12	387.136	12	364.393	12	**293.724**	324.536	339.838	18.491
		7571	11	11	650.616	11	557.356	11	382.062	11	**284.831**	289.786	296.525	4.655
		8412	10	10	1102.570	10	610.884	10	568.778	10	**424.168**	461.866	492.801	31.882
		8898	9	9	149.370	9	128.593	9	136.280	9	**123.607**	124.127	124.575	0.412
		10816	8	8	2387.010	8	1941.412	8	1884.977	8	**1864.942**	1870.116	1878.088	5.251
Arcus2	111	5755	27	27	312.585	27	300.545	27	291.041	27	**265.591**	277.055	288.723	8.228
		8847	18	18	650.651	18	460.051	18	386.052	18	**195.397**	213.506	231.029	17.293
		10027	16	16	923.228	16	538.885	16	398.996	16	**110.054**	126.653	164.775	22.778
		10743	15	15	1138.430	15	690.805	15	562.290	15	**195.761**	265.127	324.049	63.964
		11378	14	14	988.740	14	395.408	14	412.568	14	**154.143**	261.000	302.746	62.074
		17067	9	9	492.400	9	186.276	9	133.688	9	**43.525**	52.492	60.630	6.502

Tablo 4: BUMHD problemlerinin *enk DI* amaç fonksiyonuyla çözümü sonucunda elde edilen sonuçlar.

Problem	**Görev**	**C**	**Opt m**	**MRMOSA**		**MOHIH**		**DEA**				
	sayısı			**m**	**DI**	**m**	**DI**	**m**	**eniyi**	**ort**	**enkötü**	**std sapma**
Merten	7	6	6	6	**1.354**	6	**1.354**	6	**1.354**	1.354	1.354	0.000
		7	5	5	**1.414**	5	**1.414**	5	**1.414**	1.414	1.414	0.000
		8	5	5	**1.414**	5	**1.414**	5	**1.414**	1.414	1.414	0.000
		10	3	3	**0.577**	3	**0.577**	3	**0.577**	0.577	0.577	0.000
		15	2	2	**0.707**	2	**0.707**	2	**0.707**	0.707	0.707	0.000
		18	2	2	**0.707**	2	**0.707**	2	**0.707**	0.707	0.707	0.000
Jaeschke	9	6	8	8	**1.541**	8	**1.541**	8	**1.541**	1.541	1.541	0.000
		7	7	7	**2.000**	7	**2.000**	7	**2.000**	2.000	2.000	0.000
		8	6	6	**2.345**	6	**2.345**	6	**2.345**	2.345	2.345	0.000
		10	4	4	**0.866**	4	**0.866**	4	**0.866**	0.866	0.866	0.000
		18	3	3	**6.350**	3	**6.350**	3	**6.350**	6.350	6.350	0.000
Jackson	11	7	7	7	**0.845**	7	**0.845**	7	**0.845**	0.845	0.845	0.000
		9	6	6	1.732	6	**1.528**	6	**1.528**	1.609	1.732	0.112
		10	5	5	**0.894**	5	**0.894**	5	**0.894**	0.894	0.894	0.000
		13	4	4	**0.707**	4	**0.707**	4	**0.707**	0.707	0.707	0.000
		14	4	4	**0.707**	4	**0.707**	4	**0.707**	0.707	0.707	0.000
		21	3	3	**4.082**	3	**4.082**	3	**4.082**	4.082	4.082	0.000
Mitchell	21	14	8	8	**0.930**	8	**0.930**	8	0.935	0.935	0.935	0.000
		15	8	8	**0.930**	8	**0.930**	8	0.935	0.935	0.935	0.000
		21	5	5	**0.000**	5	**0.000**	5	**0.000**	0.000	0.000	0.000
Heskiaoff	28	138	8	8	4.821	8	4.359	8	**2.345**	2.345	2.345	0.000
		205	5	5	**0.447**	5	**0.447**	5	**0.447**	0.447	0.447	0.000
		216	5	5	4.098	5	1.549	5	**0.447**	0.447	0.447	0.000
		256	4	4	**0.000**	4	**0.000**	4	**0.000**	0.000	0.000	0.000
		324	4	4	73.880	4	37.356	4	**32.810**	32.810	32.810	0.000
		342	3	3	**0.816**	3	**0.816**	3	**0.816**	0.816	0.816	0.000
Sawyer	30	25	14	14	2.203	14	2.138	14	**2.035**	2.035	2.035	0.000
		27	13	13	1.300	13	**1.240**	13	**1.240**	1.240	1.240	0.000
		30	11	11	0.852	11	**0.739**	11	**0.739**	0.739	0.739	0.000
		36	10	10	1.949	10	1.844	10	**0.775**	0.799	0.894	0.054
		41	8	8	**0.707**	8	**0.707**	8	**0.707**	0.707	0.707	0.000
		54	6	6	**0.000**	6	**0.000**	6	**0.000**	0.000	0.000	0.000
		75	5	5	9.316	5	3.000	5	**1.414**	1.414	1.414	0.000
Kil.&Wes.	45	57	10	10	**0.894**	10	**0.894**	10	**0.894**	0.894	0.894	0.000
		79	7	7	**0.377**	7	**0.377**	7	**0.377**	0.377	0.377	0.000
		92	6	6	**0.000**	6	**0.000**	6	**0.000**	0.000	0.000	0.000
		110	6	6	23.544	6	13.152	6	**11.180**	14.705	15.801	2.005
		138	4	4	**0.000**	4	**0.000**	4	**0.000**	0.000	0.000	0.000
		184	3	3	**0.000**	3	**0.000**	3	**0.000**	0.000	0.000	0.000
Tonge	70	176	21	21	8.423	21	7.390	21	**5.085**	5.736	6.539	0.596
		364	10	10	4.427	10	4.939	10	**1.265**	1.397	1.549	0.105
		410	9	9	15.118	9	5.990	9	**1.202**	2.356	2.749	0.650
		468	8	8	35.114	8	9.300	8	**1.581**	1.871	2.739	0.501
		527	7	7	24.799	7	7.662	7	**1.890**	2.165	2.976	0.462
Arcus1	83	5048	16	16	236.650	16	178.386	16	**91.314**	107.784	124.731	11.839
		5853	14	14	293.290	14	142.045	14	**52.943**	65.167	70.910	7.048
		6842	12	12	516.470	12	212.473	12	**84.270**	98.389	122.463	18.861
		7571	11	11	650.616	11	378.864	11	**211.407**	232.810	264.402	19.361
		8412	10	10	1102.570	10	533.293	10	**406.648**	426.239	439.181	13.621
		8898	9	9	149.370	9	83.227	9	**27.618**	30.309	37.171	3.889
		10816	8	8	2301.090	8	1920.586	8	**1192.999**	1357.914	1553.792	176.207
Arcus2	111	5755	27	27	220.815	27	207.466	27	**137.813**	144.854	155.777	6.613
		8847	18	18	568.421	18	446.884	18	**137.618**	172.508	201.230	31.405
		10027	16	16	740.465	16	468.713	16	**182.682**	227.770	254.314	28.120
		10743	15	15	895.449	15	663.010	15	**271.752**	301.302	342.373	28.417
		11378	14	14	580.402	14	359.408	14	**187.812**	223.088	250.654	24.489
		17067	9	9	383.250	9	131.654	9	**33.612**	44.527	54.750	7.691

Tabloların ilk sütununda problem ismi, ikinci sütununda görev sayısı, üçüncü sütununda çevrim süresi ve dördüncü sütununda ilgili problemin en iyi istasyon sayısı verilmiştir. Sonraki sütunlarda karşılaştırma yapılan ve önerilen çözüm yöntemleri ile elde edilen çözümlere ait istasyon sayısı ve ilgilenilen performans ölçütleri sunulmuştur. Her bir test problemi için ilgili performans ölçütüne (Tablo 3 ve 4'te DI, Tablo 5 ve 6'da V, Tablo 7 ve 8'de hem DI hem de V) göre eniyi sonuçlar koyu

olarak gösterilmiştir. Tablo 3 ila Tablo 6'nın son üç sütununda her bir test problemi için yapılan 5 tekrarın ortalaması, en kötü değeri ve standart sapması verilmiştir.

Tablo 3'te basit düz montaj hattı dengeleme probleminin *enk DI* amaç fonksiyonuyla çözümü sonucunda elde edilen sonuçlar sunulmuştur. Önerilen algoritma ile tüm test problemleri için eniyi istasyon sayılarına ulaşılmıştır. Düzgünlük indeksi açısından ise Tablo 3'teki problemlerin %40'ında literatürden daha iyi sonuçların elde edildiği görülmektedir. Diğer test problemlerinde bilinen eniyi sonuçlara erişilmiştir. Algoritmanın başarısı özellikle büyük boyutlu problemlerde ortaya çıkmaktadır.

Tablo 5: BDMHD problemlerinin *enk V* amaç fonksiyonuyla çözümü sonucunda elde edilen sonuçlar

Problem	Görev sayısı	C	Opt m	MOGA		MOHIH		GP		DEA				
				m	V	m	V	m	V	m	V			
											eniyi	ort	enkötü	std sapma
Mitchell	21	14	8	8	**0.042**	8	**0.042**	8	**0.042**	8	**0.042**	0.042	0.042	0.000
		15	8	8	**0.090**	8	**0.090**	8	**0.090**	8	**0.090**	0.090	0.090	0.000
		21	5	5	**0.000**	5	**0.000**	5	**0.000**	5	**0.000**	0.000	0.000	0.000
Heskiaoff	28	138	8	8	0.022	8	0.022	8	0.022	8	**0.020**	0.020	0.020	0.000
		205	5	5	**0.001**	5	**0.001**	5	**0.001**	5	**0.001**	0.011	0.049	0.021
		324	4	4	**0.153**	4	**0.153**	4	**0.153**	4	**0.153**	0.153	0.153	0.000
Sawyer	30	27	13	13	0.035	13	**0.031**	13	**0.031**	13	**0.031**	0.031	0.031	0.000
		33	11	11	0.046	11	0.044	11	0.044	11	**0.038**	0.040	0.043	0.002
		54	7	7	**0.030**	7	**0.030**	7	**0.030**	7	**0.030**	0.030	0.030	0.000
Kill.&Wes	45	79	7	7	**0.004**	7	**0.004**	7	**0.004**	7	**0.004**	0.026	0.034	0.012
		92	6	6	**0.000**	6	**0.000**	6	**0.000**	6	**0.000**	0.000	0.000	0.000
		184	3	3	**0.000**	3	**0.000**	3	**0.000**	3	**0.000**	0.000	0.000	0.000
Tonge	70	176	21	21	0.031	21	0.028	21	0.024	21	**0.017**	0.018	0.020	0.001
		364	10	10	0.039	10	0.014	10	0.010	10	**0.005**	0.006	0.006	0.000
		468	8	8	0.014	8	0.013	8	0.016	8	**0.008**	0.008	0.009	0.000
Arcus1	83	5853	14	14	0.019	14	0.012	14	0.017	14	**0.007**	0.008	0.009	0.001
		6842	12	12	0.043	12	0.042	12	0.043	12	**0.041**	0.041	0.042	0.000
		8412	10	10	0.038	10	0.050	10	0.050	10	**0.034**	0.037	0.038	0.001
		10816	8	8	0.160	8	0.160	8	**0.159**	8	**0.159**	0.159	0.159	0.000
Arcus2	111	5755	27	27	0.044	27	**0.042**	27	**0.042**	27	**0.042**	0.043	0.044	0.001
		10027	16	16	0.030	16	0.030	16	0.027	16	**0.007**	0.008	0.009	0.001
		10743	15	15	0.035	15	0.035	15	0.024	15	**0.012**	0.014	0.016	0.001
		17067	9	9	0.006	9	0.006	9	0.003	9	**0.001**	0.002	0.002	0.000

Tablo 6: BUMHD problemlerinin *enk V* amaç fonksiyonuyla çözümü sonucunda elde edilen sonuçlar.

Problem	Görev sayısı	C	Opt m	MOGA		MOHIH		DEA				
				m	V	m	V	m	V			
									eniyi	ort	enkötü	std sapma
Mitchell	21	14	8	8	**0.023**	8	**0.023**	8	**0.023**	0.023	0.023	0.000
		15	8	8	**0.023**	8	**0.023**	8	**0.023**	0.023	0.023	0.000
		21	5	5	**0.000**	5	**0.000**	5	**0.000**	0.000	0.000	0.000
Heskiaoff	28	138	8	8	**0.007**	8	0.013	8	0.009	0.010	0.011	0.001
		205	5	5	**0.001**	5	**0.001**	5	**0.001**	0.001	0.001	0.000
		324	4	4	**0.062**	4	0.108	4	0.088	0.088	0.088	0.000
Sawyer	30	27	13	13	**0.023**	13	**0.023**	13	**0.023**	0.023	0.023	0.000
		33	10	10	**0.014**	10	**0.014**	10	**0.014**	0.014	0.014	0.000
		54	6	6	**0.000**	6	**0.000**	6	**0.000**	0.000	0.000	0.000
Kil.&Wes.	45	79	7	7	**0.004**	7	**0.004**	7	**0.004**	0.004	0.004	0.000
		92	6	6	**0.000**	6	**0.000**	6	**0.000**	0.000	0.000	0.000
		184	3	3	**0.000**	3	**0.000**	3	**0.000**	0.000	0.000	0.000
Tonge	70	176	21	21	0.029	21	0.021	21	**0.015**	0.016	0.019	0.002
		364	10	10	0.009	10	0.006	10	**0.002**	0.002	0.003	0.001
		468	8	8	0.004	8	0.013	8	**0.002**	0.003	0.004	0.001
Arcus	83	5853	14	14	0.014	14	0.012	14	**0.005**	0.006	0.007	0.001
		6842	12	12	0.019	12	0.018	12	**0.009**	0.010	0.011	0.001
		8412	10	10	0.038	10	0.050	10	**0.030**	0.034	0.036	0.002
		10816	8	8	0.089	8	0.157	8	**0.085**	0.111	0.128	0.020
Arcus	111	5755	27	27	0.019	27	0.018	27	**0.012**	0.013	0.013	0.000
		10027	16	16	0.029	16	0.029	16	**0.011**	0.014	0.015	0.002
		10743	15	15	0.031	15	0.032	15	**0.014**	0.016	0.017	0.001
		17067	9	9	0.004	9	0.004	9	**0.001**	0.001	0.002	0.000

Tablo 7: BDMHD problemlerinin *enk V* ve *enk DI* amaç fonksiyonlarıyla çözümü sonucunda elde edilen sonuçlar.

Problem	Görev sayısı	C	Opt m	MOHIH			GP			DEA {f=enk V}			DEA {f=enk DI}		
				m	DI	V	m	DI	V	m	DI	V	m	DI	V
Mansoor	11	48	4	4	**2.500**	**0.037**	4	**2.500**	**0.037**	4	**2.500**	**0.037**	4	**2.500**	**0.037**
		62	3	3	**0.577**	**0.007**	3	**0.577**	**0.007**	3	**0.577**	**0.007**	3	**0.577**	**0.007**
		94	2	2	**0.707**	**0.005**	2	**0.707**	**0.005**	2	**0.707**	**0.005**	2	**0.707**	**0.005**
Rosenberg	25	16	8	8	**0.791**	**0.043**	8	**0.791**	**0.043**	8	**0.791**	**0.043**	8	**0.791**	**0.043**
		21	6	6	**0.408**	**0.017**	6	**0.408**	**0.017**	6	**0.408**	**0.017**	6	**0.408**	**0.017**
		32	4	4	**1.118**	**0.025**	4	**1.118**	**0.025**	4	**1.118**	**0.025**	4	**1.118**	**0.025**
Buxey	29	30	12	12	2.309	0.037	12	2.380	0.044	12	**2.273**	**0.037**	12	**2.273**	**0.037**
		41	8	8	**0.707**	**0.012**	8	**0.707**	**0.012**	8	**0.707**	**0.012**	8	**0.707**	**0.012**
		54	7	7	4.629	0.046	7	4.720	0.058	7	**2.267**	**0.031**	7	2.828	0.047
Lutz 1	32	1572	10	10	**134.271**	**0.048**	10	**134.271**	**0.048**	10	**134.271**	**0.048**	10	**134.271**	**0.048**
		2020	8	8	**110.644**	**0.032**	8	**110.644**	**0.032**	8	**110.644**	**0.032**	8	**110.644**	**0.032**
		2828	6	6	**449.621**	**0.148**	6	**449.621**	**0.148**	6	551.674	0.148	6	449.621	0.154
Gunther	35	41	14	14	**10.660**	**0.220**	14	**10.660**	**0.220**	14	11.132	0.220	14	**10.660**	**0.220**
		61	9	9	9.214	0.089	9	**9.140**	**0.089**	9	**9.140**	**0.089**	9	**9.140**	**0.089**
		81	7	7	12.638	0.142	7	**12.638**	**0.103**	7	13.737	0.103	7	**12.603**	**0.147**
Hahn	53	2004	8	8	362.975	0.135	8	**342.121**	**0.135**	8	364.446	0.135	8	**328.402**	**0.136**
		2338	7	7	**531.709**	**0.177**	7	**531.709**	**0.177**	7	**531.709**	**0.177**	7	**531.709**	**0.177**
		3507	5	5	**876.707**	**0.161**	5	**876.707**	**0.161**	5	891.228	0.161	5	**876.707**	**0.161**
Wee-Mag	75	30	62	62	6.287	0.079	62	**5.759**	**0.077**	62	6.287	0.079	62	6.287	0.079
		40	60	60	14.597	0.104	60	**13.640**	**0.104**	60	**15.570**	**0.103**	60	**12.675**	**0.109**
		52	31	31	3.121	0.035	31	3.292	0.038	31	2.962	0.026	31	**2.951**	**0.026**
Lutz 2	89	15	34	34	1.361	0.076	34	1.339	0.074	34	1.317	0.073	34	**1.295**	**0.071**
		19	26	27	1.333	0.044	27	1.333	0.044	27	**1.186**	**0.030**	27	**1.186**	**0.030**
		21	24	24	1.242	0.041	24	1.242	0.041	24	**1.172**	**0.041**	24	**1.172**	**0.041**

Tablo 8: BUMHD problemlerinin *enk V* ve *enk DI* amaç fonksiyonlarıyla çözümü sonucunda elde edilen sonuçlar.

Problem	Görev sayısı	C	Opt m	MOHIH			DEA {f=enk V}			DEA {f=enk DI}		
				m	DI	V	m	DI	V	m	DI	V
Mansoor	11	48	4	4	**2.061**	**0.022**	4	**2.061**	**0.022**	4	**2.061**	**0.022**
		62	3	3	**0.577**	**0.007**	3	**0.577**	**0.007**	3	**0.577**	**0.007**
		94	2	2	**0.707**	**0.005**	2	**0.707**	**0.005**	2	**0.707**	**0.005**
Rosenberg	25	16	8	8	**0.612**	**0.030**	8	**0.612**	**0.030**	8	**0.612**	**0.030**
		21	6	6	**0.408**	**0.017**	6	**0.408**	**0.017**	6	**0.408**	**0.017**
		32	4	4	**0.866**	**0.013**	4	**0.866**	**0.013**	4	**0.866**	**0.013**
Buxey	29	30	11	11	0.852	0.021	11	**0.739**	**0.017**	11	**0.739**	**0.017**
		41	8	8	**0.707**	**0.012**	8	**0.707**	**0.012**	8	**0.707**	**0.012**
		54	6	6	**0.000**	**0.000**	6	**0.000**	**0.000**	6	**0.000**	**0.000**
Lutz1	32	1572	10	10	38.771	0.009	10	**31.023**	0.009	10	**31.023**	0.009
		2020	8	8	58.974	0.014	8	**18.055**	**0.006**	8	**18.055**	**0.006**
		2828	6	6	210.101	0.057	6	**181.207**	**0.049**	6	**181.207**	**0.049**
Gunther	35	41	12	12	**1.040**	**0.017**	12	**1.040**	**0.017**	12	**1.040**	**0.017**
		61	8	8	**0.790**	**0.007**	8	**0.790**	**0.007**	8	**0.790**	**0.007**
		81	6	6	**0.707**	**0.006**	6	**0.707**	**0.006**	6	**0.707**	**0.006**
Hahn	53	2004	8	8	268.995	0.095	8	**211.314**	**0.063**	8	**211.314**	**0.063**
		2338	7	7	329.367	0.085	7	**293.198**	**0.060**	7	**293.198**	**0.060**
		3507	5	5	**788.605**	**0.161**	5	891.228	0.161	5	**752.210**	**0.163**
Wee-Mag	75	30	62	62	6.210	0.072	62	**6.179**	**0.069**	62	**6.179**	**0.069**
		40	60	60	**12.622**	**0.096**	60	**15.474**	**0.093**	60	**10.615**	**0.100**
		52	31	31	3.121	0.029	31	2.940	0.025	31	**2.874**	**0.022**
Lutz2	89	15	33	33	**0.550**	**0.030**	33	**0.550**	**0.030**	33	**0.550**	**0.030**
		19	26	26	0.650	0.028	26	**0.588**	**0.025**	26	**0.588**	**0.025**
		21	24	24	**0.889**	**0.019**	24	**0.889**	**0.019**	24	**0.889**	**0.019**

Tablo 4'te basit U-tipi montaj hattı dengeleme probleminin *enk DI* amaç fonksiyonuyla çözümü sonucunda elde edilen sonuçlar sunulmuştur. Tüm test problemlerinde eniyi istasyon sayılarına ve problemlerin %44'ünde literatürden daha iyi düzgünlük indeksine sahip sonuçlara ulaşılmıştır. İki problem dışındaki diğer test problemlerinde bilinen eniyi sonuçlara erişilmiştir.

Tablo 5 ve 6'da sırasıyla düz ve U-tipi basit montaj hattı dengeleme problemlerinin *enk V* amaç fonksiyonuyla çözümü sonucunda elde edilen sonuçlar sunulmuştur. Her iki tabloda da tüm test problemlerinde eniyi istasyon sayılarına ulaşılmış ve problemlerin %48'inde literatürdeki sonuçlar iyileştirilmiştir.

Tablo 7 ve 8'de sırasıyla düz ve U-tipi basit montaj hattı dengeleme problemlerinin *enk V* ve *enk DI* amaç

fonksiyonuyla ayrı ayrı çözümü sonucunda elde edilen sonuçlar sunulmuştur. Tablo 7'den düz montaj hattı dengeleme problemlerinden çevrim süresi 19 olan Lutz2 problemi dışındaki tüm veri setleri ve çevrim sürelerinde en iyi istasyon sayılarına ulaşıldığı görülmektedir. U-tipi problemlerde ise tüm test problemleri için en iyi istasyon sayılarına ulaşılmıştır. Karşılaştırma yapılan MOHIH [36] ve GP [31] yöntemleri istasyon sayıları açısından DEA ile aynı performansı gösterdiğinden yöntemler, düzgünlük indeksi (DI) ve işyükü değişimi (V) ölçütlerini dikkate alarak iki ölçütlü olarak karşılaştırılmıştır. Tablo 7 ve 8'de hem koyu hem de altı çizili değerler birbirine baskın olmayan çözümleri göstermektedir. Tablo 7'deki 24 problemin 16'sında önceki çalışmalarla aynı sonuçlara ulaşılmış, 3'ünde ulaşılamamış, 2'sinde ise her iki ölçüt için de daha iyi çözüm elde edilmiştir. Tablo 8'de ise 24 problemin 21'inde önceki çalışmalarla aynı sonuçlara ulaşılmış, 1'inde her iki ölçüt için de daha iyi çözüm elde edilmiştir.

5 Sonuç ve öneriler

Rekabet açısından eldeki kaynakların en iyi şekilde değerlendirilmesinin bir zorunluluk olduğu günümüzde, montaj hatlarının en iyi şekilde dengelenmesi, işletmelerin kapasitelerini etkin kullanabilmeleri açısından kritik önemdedir. Seri üretimin temelini oluşturan geleneksel düz montaj hatları zamanla müşteri beklentileri, rekabet faktörleri ve teknolojik ilerlemelere bağlı olarak tasarım açısından değişik şekillerde kullanılmıştır. Bunlardan birisi olan U-tipi montaj hatlarının kullanımı, Tam Zamanında Üretim (Just in Time) ilkesinin uygulanmasıyla birlikte giderek artmıştır.

Bu çalışmada düz ve U-tipi basit montaj hattı dengeleme problemlerinin çözümü için geliştirilen diferansiyel evrim algoritmasının literatürdekilerden farkı, kromozom yapısının sağlamlığı olmaktadır. Gerçekten de, kullanılan kromozom yapısı hem öncelik ilişkilerine ağırlık verebilmekte, hem tamir gerektirmemekte hem de dönüşümde rassallığa yer vermediği için kendisi ile atama şekli arasında deterministik bir haritalama sağlamaktadır. Önerilen algoritmanın çözüm başarısı, literatürde yaygın olarak kullanılan çok sayıda test problemi kullanılarak gerçekleştirilen deneyler ile değerlendirilmiştir. Çözülen 208 problemden sadece bir tanesinde eniyi çözüme (istasyon sayısı) ulaşılamamış, diğerlerinde ise oldukça kısa sürelerde eniyi çözümlere ulaşılmıştır. Sonuçlar düzgünlük indeksi (DI) ve işyükü değişimi (V) açısından karşılaştırıldığında ise diferansiyel evrim algoritması ile literatürdeki sonuçlardan daha iyi sonuçlara erişilebildiği görülmüştür. Algoritmanın başarısı özellikle büyük boyutlu problemlerde ortaya çıkmaktadır.

Çalışmanın devamında, en iyi performansı verecek algoritma parametrelerinin belirlenmesi için deney tasarımı yapılması düşünülmektedir. Ayrıca önerilen algoritma, karma montaj hattı dengeleme problemi gibi daha karmaşık hat dengeleme problemlerinin çözümünde kullanılabilir.

6 Kaynaklar

[1] Erkut H, Baskak M. *Stratejiden Uygulamaya Tesis Tasarımı*. İstanbul, Türkiye, İrfan Yayınevi, 2003.

[2] Kara Y. U-Tipi Montaj Hattı Dengeleme Problemleri için Yeni Modeller ve Otomotiv Yan Sanayiinde Bir Uygulama. Doktora Tezi, Selçuk Üniversitesi, Konya, Türkiye, 2004.

[3] Ajenblit DA, Wainwright RL. "Applying genetic algorithms to the u-shaped assembly line balancing problem". *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*. Anchorage, Alaska, USA, 04-09 May 1998.

[4] Salveson ME. "The assembly line balancing problem". *Journal of Industrial Engineering*, 6(3), 18-25. 1955.

[5] Bowman EH. "Assembly line balancing by linear programming". *Operations Research*, 8(3), 385-389, 1960.

[6] Klein M. "On assembly line balancing". *Operations Research*, 11, 274-281, 1963.

[7] Patterson JH, Albracht JJ. "Assembly-line balancing: Zero-one programming with fibonacci search". *Operations Research*, 23(1), 166-172, 1975.

[8] Talbot FB. Patterson JH. "An integer programming algorithm with network cuts for solving the assembly line balancing problem". *Management Science*, 30(1), 85-99, 1984.

[9] Jackson JR. "A computing procedure for a line balancing with a precedence matrix". *Management Science*, 2, 261-272, 1956.

[10] Held M, Karp RM, Shareshian R. "Assembly line balancing-dynamic programming with precedence constraints". *Operations Research*, 11, 442-459, 1963.

[11] Schrage L, Baker KR. "Dynamic programming solution of sequencing problems with precedence constraints". *Operations Research*, 26, 444-449. 1978.

[12] Johnson RV. "Assembly line balancing algorithms: Computational comparisons". *International Journal of Production Research*, 19, 277-287, 1981.

[13] Liu SB, Ng KM, Ong HL. "Branch-and-bound algorithms for simple assembly line balancing problem". *International Journal of Advanced Manufacturing Technology*, 36, 169-177, 2008.

[14] Dar-El EM. "MALB-a heuristic technique for balancing large scale single-model assembly lines". *AIIE Transactions*, 5, 343-356, 1973.

[15] Dar-El EM, Rubinovitch Y. "MUST-A multiple solutions technique for balancing single model assembly lines". *Management Science*, 25, 1105-1114, 1979.

[16] Baybars I. "An efficient heuristic method for the simple assembly line balancing problem". *International Journal of Production Research*, 24, 149-166, 1986.

[17] Tonge FM. "Assembly line balancing using probabilistic combinations of heuristics". *Management Science*, 11, 727-735. 1965.

[18] Moodie CL, Young HH. "A heuristic method of assembly line balancing for assumptions of constant or variable work element times". *Journal of Industrial Engineering*, 16, 23-29, 1965.

[19] Nevins AJ. "Assembly line balancing using best bud search". *Management Science,* 18, 529-539, 1972.

[20] Kim YJ, Kim YK, Cho Y. "A heuristic-based genetic algorithm for workload smoothing in assembly lines". *Computers and Operations Research*, 25(2), 99-111. 1998.

[21] Chan KCC, Hui PCL, Yeung KW, Ng FSF. "Handling the assembly line balancing problem in the clothing industry using a genetic algorithm". *International Journal of Clothing Science and Technology*, 10(1), 21-37, 1998.

[22] Sabuncuoğlu I, Erel E, Tanyer M. "Assembly line balancing using genetic algorithms". *Journal of Intelligent Manufacturing*, 11, 295-310, 2000.

[23] Ponnambalam SG, Aravindan P, Naidu GM. "A multiobjective genetic algorithm for solving assembly line balancing problem". *International Journal of Advanced Manufacturing Technology*, 16, 341-352, 2000.
[24] Goncalves JF, Almeida JR. "A hybrid genetic algorithm for assembly line balancing". *Journal of Heuristics*, 8, 629-642. 2002.
[25] Hwang RK, Katayama H, Gen M. "U-Shaped assembly line balancing problem with genetic algorithm". *International Journal of Production Research*, 46(16), 4637-4649, 2008.
[26] Scholl A, Voß S. "Simple assembly line balancing-heuristic approaches". *Journal of Heuristics*, 2, 217-244, 1996.
[27] Chiang WC. "The application of a tabu search metaheuristic to the assembly line balancing problem". *Annals of Operations Research*, 77, 209-227, 1998.
[28] Lapierre SD, Ruiz A, Soriano P. "Balancing assembly lines with tabu search". *European Journal of Operational Research,* 168, 826-837, 2006.
[29] Bautista J, Pereira J. "Ant algorithms for a time and space constrained assembly line balancing problem". *European Journal of Operational Research,* 177, 2016-2032, 2007.
[30] McMullen PR, Tarasewich P. "Using ant techniques to solve the assembly line balancing problem". *IIE Transactions,* 35, 605-617, 2003.
[31] Baykasoğlu A, Özbakır L. "Discovering task assignment rules for assembly line balancing via genetic programming". *International Journal of Advanced Manufacturing Technology,* 76, 417-434, 2015.
[32] Heinrici A. *A Comparison Between Simulated Annealing and Tabu Search With an Example From the Production Planning.* Editor: Dyckhoff H. Operations research proceedings. 498-503, Berlin, Germany, Springer, 1994.
[33] Suresh G, Sahu S. "Stochastic assembly line balancing using simulated annealing". *International Journal of Production Research*, 32, 1801-1810, 1994.
[34] McMullen PR, Frazier GV. "Using simulated annealing to solve a multi objective assembly line balancing problem with parallel workstations". *International Journal of Production Research*, 36, 2717-2741, 1998.
[35] Baykasoğlu A. "Multi-Rule multi-objective simulated annealing algorithm for straight and U type assembly line balancing problems". *Journal of Intelligent Manufacturing*, 17, 217-232, 2006.
[36] Özcan U, Toklu B. "A new hybrid improvement heuristic approach to simple straight and u-type assembly line balancing problems". *Journal of Intelligent Manufacturing*, 20(1), 123-136, 2009.
[37] Miltenburg GJ, Wijngaard J. "The u-line balancing problem". *Management Science,* 40(10), 1378-1388, 1994.
[38] Urban TL. "Note Optimal balancing of U-shaped assembly lines". *Management Science,* 44(5), 738-741, 1998.
[39] Scholl A, Klein R. "ULINO: Optimally balancing u-shaped JIT assembly lines". *International Journal of Production Research*, 37(4), 721-736, 1999.
[40] Erel E, Sabuncuoglu I, Aksu BA. "Balancing of U-type assembly systems using simulated annealing". *International Journal of Production Research,* 39, 3003-3015, 2001.
[41] Aase GR, Schniederjans MJ, Olson JR. "U-OPT: An analysis of exact U-Shaped line balancing procedures". *International Journal of Production Research,* 41, 4185-4210, 2003.
[42] Gökcen H, Ağpak K, Gencer C, Kizilkaya E. "A shortest route formulation of simple u-type assembly line balancing problem". *Applied Mathematical Modelling*, 29, 373-380, 2005.
[43] Gökcen H, Ağpak K. "A goal programming approach to simple u-line balancing problem". *European Journal of Operational Research,* 171, 577-585, 2006.
[44] Toklu B, Özcan U. "A fuzzy goal programming model for the simple u-line balancing problem with multiple objectives". *Engineering Optimization,* 40(3), 191-204, 2008.
[45] Storn R, Price K. "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces". *Journal of Global Optimization*, 11, 341-359, 1997.
[46] Nearchou AC. "Balancing large assembly lines by a new heuristic based on differential evolution method". *International Journal of Advanced Manufacturing Technology*, 34, 1016-1029, 2007.
[47] Mozdgir A, Mahdavi I, Badeleh IS, Solimanpur M. "Using the Taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing". *Mathematical and Computer Modelling,* 57, 137-151, 2013.
[48] Nearchou AC. "Multi-objective balancing of assembly lines by population heuristics". *International Journal of Production Research,* 46(8), 2275-2297, 2008.
[49] Nourmohammadi A, Zandieh M. "Assembly line balancing by a new multi-objective differential evolution algorithm based on TOPSIS". *International Journal of Production Research*, 49(10), 2833-2855, 2011.
[50] Zhang H, Yan Q, Liu Y, Jiang Z. "An integer-coded differential evolution algorithm for simple assembly line balancing problem of type 2". *Assembly Automation,* 36(3), 246-261, 2016
[51] Nearchou AC. "A differential evolution algorithm for simple assembly line balancing". *IFAC 16th Triennial World Congress*, Prague, Czech republic, 4-8 July 2005.
[52] Pitakaso R. "Differential evolution algorithm for simple assembly line balancing type 1 (SALBP-1)". *Journal of Industrial and Production Engineering,* 32(2), 104-114, 2015.
[53] Pitakaso R, Sethanan K. "Modified differential evolution algorithm for simple assembly line balancing with a limit on the number of machine types". *Engineering Optimization*, 48(2), 253-271, 2016.
[54] Rönkkönen JI, Kukkonen S, Price K. "Real-Parameter optimization with differential evolution". *IEEE Congress on Evolutionary Computation,* Edinburgh, Scotland, 2-5 September 2005.