

## Kübik hücre oluşturma problemi için arı algoritması Bees algorithm for cubic cell formation problem

Hakan GÖKDAŞ<sup>1</sup> , Feriştah ÖZÇELİK<sup>2\*</sup> 

<sup>1,2</sup>Endüstri Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Eskişehir Osmangazi Üniversitesi, Eskişehir, Türkiye.  
hakangoktas@gmail.com, fdurmaz@ogu.edu.tr

Geliş Tarihi/Received: 19.06.2019  
Kabul Tarihi/Accepted: 01.11.2019

Düzeltilme Tarihi/Revision: 31.10.2019

doi: 10.5505/pajes.2019.85125  
Araştırma Makalesi/Research Article

### Öz

Hücre oluşturma problemi ile ilgili çalışmaların çoğunluğunda sadece parça ve makine atanması ele alınmaktadır. İşlerin makinelerde yapılması sırasındaki önemli rolü nedeniyle işçilerin hücelere atanması, hücresel üretim sisteminden beklenen faydanın elde edilmesinde oldukça kritik bir faktör olmaktadır. İşçilerin ve makinelerin parça işlemedeki yeteneklerini dikkate almak, kaynakların verimliliğini ve esnekliği artırmada oldukça önemlidir. Bu çalışmada, üç boyutlu parça-makine-işçi temas matrisini dikkate alan kübik hücre oluşturma probleminin çözümü için arı algoritması geliştirilmiştir. Geliştirilen algoritma, hücre sayısının önceden bilindiği ve sabit olduğu literatürdeki diğer çalışmalardan farklı olarak, yeni hücre sayısını belirleyebilmektedir. Arı algoritmasının performansı, literatürden alınan problemler kullanılarak test edilmiştir. Elde edilen sonuçlar, çözülen 11 problemin dokuzunda önerilen algoritmanın bilinen eniyi çözümlere ulaştığını ve kalan ikisinde de daha iyi çözümler bularak başarılı olduğunu göstermiştir. Hücre sayısının bilinmediği durumda ise problemlerin %100'ünde önerilen algoritma ile daha iyi çözümlere erişilmiştir.

**Anahtar kelimeler:** Hücresel üretim, Kübik hücre oluşturma problemi, Arı algoritması.

### Abstract

In the literature, most of the studies on cell formation problem only consider the assignment of parts and machines into the cells. Since workers have important role in doing jobs on machines, assignment of workers to cells becomes a crucial factor for fully utilization of cellular manufacturing systems. Considering the abilities of workers and the machines for processing parts are important to enhance flexibility and efficiency of resources. In this study, bees algorithm is developed to solve the cubic cell formation problem that considers the parts, the machines and the workers together while forming manufacturing cells. Unlike the methods in literature that define the number of cells as a constant number, the algorithm is capable of determining the optimum number of cells. The performance of the developed algorithm is tested by using problems from literature. The results showed that the proposed algorithm is able to find the best known solutions in nine out of 11 problems and better results for the remaining problems. For the case that the number of cells is not known, the algorithm improved the solutions for 100% of the problems.

**Keywords:** Cellular manufacturing, Cubic cell formation problem, Bees algorithm.

## 1 Giriş

Günümüzün değişken piyasa koşulları göz önünde bulundurulduğunda bir firmanın ayakta kalabilmesi ve rekabet edebilmesi için ürünlerini müşteri istekleri doğrultusunda ve ekonomik olarak üretebilmesi ve zamanında teslim edebilmesi gerekmektedir. Esnek üretim sistemleri, bunu sağlayacak çözümlerden birisidir. Bu sistemlerin alt yapısı da, hücresel üretime dayanmaktadır. Bu yenilikçi üretim felsefesinin uygulanabilmesi için, öncelikle üretim hücrelerinin oluşturulması gerekmektedir.

Başarılı bir hücresel üretim sistemi uygulaması için üretim hücreleri oluşturulurken dikkate alınması gereken dört temel unsur; insan, donanım, malzeme ve işlemdir [1]. Literatürdeki çalışmalar genellikle son üç unsuru göz önüne alarak, parçaların üretimi için gerekli olan işlemler, bu işlemlerin gerçekleştirildiği makineler ve işlem sıraları, üretilen parçaların talep miktarları, üretim parti büyüklükleri, hazırlık ve işlem süreleri, alternatif rota gibi birçok özellik üzerinde durmaktadır. Oysa hücresel üretim sistemlerinin tasarımında; donanım kullanımı için “teknik beceriler” ve takım çalışmasına uyum sağlama, problem çözebilme, karar verebilme yeteneği gibi detaylandırılabilir “kişisel beceriler” gereksinimleri nedeniyle insan unsuru, diğer üç unsura ait özelliklerle birlikte dikkate alınmalıdır. Bu unsurların tamamını dikkate alan hücre

oluşturma problemi, kübik hücre oluşturma problemi olarak isimlendirilmektedir. Klasik (insan unsurunu dikkate almayan) hücre oluşturma probleminin girdisi genellikle 0-1 parça-makine temas matrisidir (iki boyutlu). Parça-makine temas matrisi  $A = [a_{ij}]$ ,  $P$  parça sayısı ve  $M$  makine sayısı iken elemanları 0 veya 1 olan  $P \times M$  boyutlu bir matristir. Bu matrisin sıfır olan elemanları  $a_{ij} = 0$ ,  $i$ . parçanın  $j$ . makinede işlemi olmadığını; bir olan elemanları da  $a_{ij} = 1$ ,  $i$ . parçanın  $j$ . makinede işlemi bulunduğunu gösterir. İki boyutlu hücre oluşturma problemi için örnek bir temas matrisi ve çözümü Şekil 1’de verilmiştir.

	M1	M2	M3	M4		M4	M2	M3	M1
P1	0	0	1	0	P3	1	1	0	0
P2	1	0	0	0	P5	1	0	1	0
P3	0	1	0	1	P1	0	0	1	0
P4	0	0	1	0	P4	0	0	1	0
P5	0	0	1	1	P2	0	0	0	1
P6	1	0	0	0	P6	0	0	0	1

Şekil 1. İki boyutlu hücre oluşturma problemi. (a): 0-1 temas matrisi, (b): Köşegenleştirilmiş matris.

Figure 1. Two dimensional cell formation problem. (a): 0-1 incidence matrix, (b): Block diagonalised matrix.

\*Yazışılan yazar/Corresponding author

Verilen üç hücreli çözümde, hücre dışında kalan 1'ler hücre dışı eleman ve hücre içindeki 0'lar boşluk olarak isimlendirilmekte ve sırasıyla hücrelerarası ve hücre içi taşımalara karşılık gelmektedir. Amaç, hücre dışı eleman ve boşluk sayılarının toplamının enküçüklenmesidir.

Kübik hücre oluşturma probleminde ise, üç boyutlu parça-makine-işçi temas matrisi  $R = [r_{imw}]$  dikkate alınmaktadır.  $P$  parça sayısı,  $M$  makine sayısı ve  $W$  işçi sayısı iken  $R$  matrisi, elemanları 0 veya 1 olan  $P \times M \times W$  boyutlu bir matristir. Matrisin elemanı  $r_{imw}$ , eğer parça  $i$ , makine  $m$ 'de işçi  $w$  ile işlenebiliyorsa 1, diğer durumda 0 değerini almaktadır. İki boyutlu probleme benzer şekilde üç boyutlu hücre oluşturma probleminde de hücre dışında kalan 1'ler hücre dışı eleman ve hücre içindeki 0'lar boşluk olarak adlandırılmaktadır. Hücre dışı elemanlar, parça, parçanın işlem göreceği makine ve parçayı ilgili makinede işleyecek işçinin aynı hücreye atanmaması durumunda ortaya çıkmaktadır. Bu durumda ya parçanın ya da işçinin başka bir hücreye taşınması söz konusu olacaktır. Bununla ilgili üç durumda karşılaşılabılır.

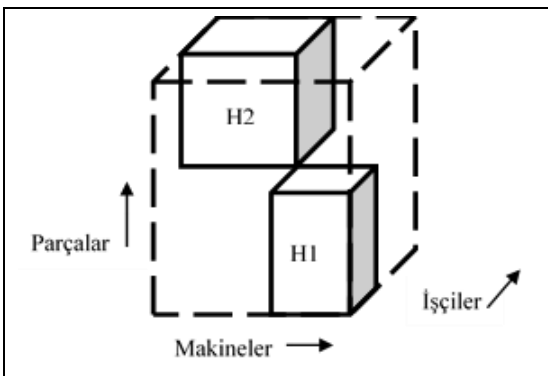
*Durum 1:* Parça ve makinenin aynı hücreye, işçinin ise onlardan farklı bir hücreye atanması.

*Durum 2:* Makine ve işçinin aynı hücreye, parçanın onlardan farklı bir hücreye atanması.

*Durum 3:* Hem parça hem de işçinin makineden farklı bir hücreye atanması.

Durum 1'de işçi, parça ve makinenin olduğu hücreye giderek parçayı işlemekte ve işçi için hücreler arası hareket gerekmektedir. Durum 2'de parça, işçi ve makinenin olduğu hücreye taşınarak işlem görmekte ve parça için hücreler arası hareket gerekmektedir. Durum 3'te ise hem parçanın hem de işçinin makinenin olduğu hücreye gitmesi gerektiğinden hem parça hem de işçi için hücreler arası hareket gerekmektedir. İlk iki durumda hücre dışı eleman sayısı 1 iken son durumda 2 olacaktır.

İşçileri de dikkate alan üç boyutlu (kübik) hücre oluşturma problemi (Şekil 2) ilk olarak Min ve Shin [2] tarafından ortaya atılmış ve işçi, parça-makine temas matrisine üçüncü faktör olarak yerleştirilmiştir.



Şekil 2. Üç boyutlu (kübik) hücre oluşturma problemi.

Figure 2. Three dimensional (cubic) cell formation problem

Li [3], tüm temas matrislerini aynı anda dikkate alan bir yöntem önermiştir. Mahdavi ve diğ. [4], işçi atama esnekliğini göz önüne alan dinamik bir ortamda hücreli üretim sistemlerinin tasarımı için bir tam sayılı matematiksel programlama modeli sunmuştur. Bu modelin amaçları, elde tutma ve bekleyen sipariş maliyetlerini, hücrelerarası taşıma maliyetlerini, makine ve yeniden düzenleme maliyetleri ve işe alma, atma ve

maaş giderlerini enküçükmektir. Mahdavi ve diğ. [5] kübik hücre oluşturma problemi için bir matematiksel model önermiştir. Nikoofarid ve Aalaei [6], makine kapasitesini, talebi ve ürün karması ile çok dönemli planlama ufkunu göz önüne alan, dinamik bir sanal ortamda üretim planlaması için bir matematiksel model önermiştir. Bu modelin başlıca kısıtları, tüm dönemlerdeki talep tatmini, makine uygunluğu, makine zamanı kapasitesi ve işçilerin uygunluğudur. Bu modelin amacı, planlama ufkunu boyunca elde tutma ve bekleyen sipariş maliyetlerini en aza indirmektir. Aaleei ve Shavazipour [7], en verimli seçeneği belirlemek için veri zarflama analizini kullanarak yeni bir matematiksel model önermiştir. Amaç, bekleyen sipariş ve hücrelerarası taşıma maliyetlerini en aza indirmektir. Bootaki ve diğ. [8], hücrelerarası taşımaların enküçüklenmesi ve parça kalite indeksinin enbüyüklenmesi amaçlarını dikkate alan iki amaçlı kübik hücre oluşturma problemini ele almıştır. Problemin çözümü için genetik algoritma ve genişletilmiş epsilon kısıt yöntemlerini melezleyen bir yaklaşım önermişlerdir. Buruk Şahin ve Alpay [9] kübik hücre oluşturma probleminin çözümü için bir genetik algoritma önermiştir.

Kübik hücre oluşturma problemini ele alan çalışmalar incelediğinde bu çalışmalarda hücre sayısının önceden bilindiği ve sabit olduğu durumun ele alındığı dikkati çekmiştir. Uygulamada tasarımcılar en iyi hücreli üretim sistemi (en iyi amaç fonksiyonu değerine sahip) tasarımına karşı gelen hücre sayısını bilmezler. O halde hücre sayısının önceden bilindiği varsayımı gerçekçi değildir [10]. İki boyutlu hücre oluşturma problemi ile ilgili çalışmalar, hücre sayısının sistemin performansı/verimliliği üzerinde etkisi olduğunu göstermektedir [11]. Bunu destekleyecek şekilde son yıllarda değişken hücre sayısını dikkate alan çalışmaların arttığı görülmüştür [12]-[14]. Ancak hücre sayısının bilinmediği durumda problem daha da zorlaşmaktadır [14]. Bu çalışmada kübik hücre oluşturma probleminin daha genel hali olan hücre sayısının bilinmediği durum ilk kez ele alınmıştır.

Karmaşık optimizasyon problemlerinin çözümünde, meta sezgisel yöntemlerin kullanımına yönelik bir eğilim vardır. Doğadan esinlenen algoritmaların yeni bir dalı olan sürü zekâsı yaklaşımı, böceklerin içgüdüsel problem çözme becerilerini kullanarak etkili meta sezgisel yöntemler geliştirebilmek için böcek davranışlarının modellenmesine odaklanmıştır [15]. Sürü zekâsına dayanan popülasyon temelli bir arama algoritması olan arı algoritmasının, genelleştirilmiş atama problemi [15],[16], montaj hattı dengeleme problemi [17],[18], iki boyutlu hücre oluşturma problemi [19] gibi bir çok farklı probleme başarıyla uygulanmış olmasına rağmen kübik hücre oluşturma probleminde uygulanmamış olması dikkat çekicidir. Bu çalışmada kübik hücre oluşturma probleminin çözümü için arı algoritması (AA) ilk kez önerilmiştir. Önerilen algoritma hem eniyi hücre sayısını hem de hücrelere atanacak parça, makine ve işçileri belirlemektedir.

Makalenin ikinci bölümünde ele alınan problemin matematiksel modeli sunulmuş, üçüncü bölümünde geliştirilen arı algoritması anlatılmış ve dördüncü bölümde de deneysel sonuçlar verilmiştir. Son bölümde ise elde edilen sonuçlar tartışılmış ve önerilerde bulunulmuştur.

## 2 Kübik hücre oluşturma problemi

Bu çalışmada ele alınan kübik hücre oluşturma probleminin matematiksel modeli, Mahdavi ve diğ. [5] tarafından önerilmiştir.

#### Varsayımlar

- Makine ve işçiler, sınırsız kapasiteye sahiptir,
- Her işçi tipinden yalnızca bir adet vardır,
- Bazı tip makinelerden birden fazla sayıda mevcuttur,
- İşçiler çeşitli makinelerde çalışabilir,
- Her parça, işlem göreceği makinelerde sadece bir işçi ile işlenecektir.

#### İndisler

$i$ : parça indisi ( $i = 1, 2, \dots, P$ )

$w$ : işçi indisi ( $w = 1, 2, \dots, W$ )

$m$ : makine indisi ( $m = 1, 2, \dots, M$ )

$k$ : hücre indisi ( $k = 1, 2, \dots, C$ )

#### Parametreler

$r_{imw}$ : parça  $i$ , işçi  $w$  ile makine  $m$ 'de işlenebiliyorsa 1; diğer durumda 0

$a_{im}$ : parça  $i$ 'nin makine  $m$ 'de işlemi varsa 1; diğer durumda 0

$N_m$ :  $m$  tipi makine sayısı

$LM_k$ :  $k$  hücresindeki makine sayısı için alt sınır

$LP_k$ :  $k$  hücresindeki parça sayısı için alt sınır

$LW_k$ :  $k$  hücresindeki işçi sayısı için alt sınır

#### Karar değişkenleri

$x_{mk}$ : makine  $m$ , hücre  $k$ 'ya atanmışsa 1; diğer durumda 0

$y_{ik}$ : parça  $i$ , hücre  $k$ 'ya atanmışsa 1; diğer durumda 0

$z_{wk}$ : işçi  $w$ , hücre  $k$ 'ya atanmışsa 1; diğer durumda 0

$d_{imwk}$ : parça  $i$ , makine  $m$  ile işçi  $w$  tarafından hücre  $k$ 'da işleniyorsa 1; diğer durumda 0

#### Matematiksel model

$$\sum_{k=1}^C Z_{wk} = 1 \quad \forall w \quad (1)$$

$$\sum_{k=1}^C y_{ik} = 1 \quad \forall i \quad (2)$$

$$\sum_{k=1}^C x_{mk} \leq N_m \quad \forall m \quad (3)$$

$$\sum_{m=1}^M X_{mk} \geq LM_k \quad \forall k \quad (4)$$

$$d_{imwk} \leq r_{imw}x_{mk} \quad \forall i, m, w, k \quad (5)$$

$$\sum_{k=1}^C \sum_{w=1}^W d_{imwk} = a_{im} \quad \forall i, m \quad (6)$$

$$\sum_{i=1}^P y_{ik} \geq LP_k \quad \forall k \quad (7)$$

$$\sum_{w=1}^W Z_{wk} \geq LW_k \quad \forall k \quad (8)$$

$$x_{mk}, y_{ik}, z_{wk}, d_{imwk} \in \{0,1\} \quad \forall i, m, w, k \quad (9)$$

kısıtları altında

$$Enk \left[ \sum_{k=1}^C \left[ \sum_{i=1}^P \sum_{m=1}^M \sum_{w=1}^W y_{ik}x_{mk}z_{wk} - \sum_{i=1}^P \sum_{m=1}^M \sum_{w=1}^W y_{ik}x_{mk}z_{wk}d_{imwk} \right] + \sum_{i=1}^P \sum_{k=1}^C \sum_{m=1}^M \sum_{w=1}^W x_{mk}(2 - y_{ik} - z_{wk})d_{imwk} \right] \quad (10)$$

Eş. (1) her işçinin yalnızca bir hücreye atanmasını sağlar. Eş. (2), her parçanın yalnızca bir hücreye atanmasını ifade eder. Eş. (3),  $m$  tipi makineden hücrelere ilgili makinenin mevcut sayısını aşmayacak kadar atanmasını garanti eder. Eş. (4), her hücreye makine sayısı alt sınırına eşit ya da daha fazla sayıda makine atanmasını sağlar. Eş. (5),  $k$  hücresinde  $m$  tipi makine yoksa  $d_{imwk} = 0$  olmasını sağlar. Eş. (6), bir hücredeki her makinede her parça işlenmesi için yalnızca bir işçinin görevlendirileceğini belirtir. Eş. (7), her hücreye parça sayısı alt sınırına eşit ya da daha fazla sayıda parça atanmasını sağlar. Eş. (8), her hücreye işçi sayısı alt sınırına eşit ya da daha fazla sayıda işçi atanmasını sağlar. Eş. (9) karar değişkenlerine ait işaret kısıtlarıdır. Amaç fonksiyonunun (10) ilk kısmı boşluk sayısını hesaplar; ikinci kısım hücre dışı eleman sayısını hesaplar. Amaç, boşluk ve hücre dışı eleman sayılarının toplamının en küçüklenmesidir.

### 3 Önerilen arı algoritması

Arı algoritması, Pham ve diğ. [20] tarafından önerilmiş olup, sürü zekasına dayanan popülasyon temelli bir arama algoritmasıdır. Bu algoritma, doğada sürü halinde hareket eden arıların besin bulmada sergilemiş oldukları davranışları temel olarak optimizasyon problemlerini çözmek için kullanılmaktadır. Arıların yiyecek aramak için gittikleri kaynaklar, algortmada çözülmek istenen problemin olası çözümlerini, kaynaklardaki nektar miktarı ise çözümün kalitesini (amaç fonksiyonu) ifade etmektedir. Arı algoritmasında, bir kolonide işçi arı, gözcü arı ve kâşif arı olmak üzere üç çeşit arı bulunmaktadır. Kâşif arılar herhangi bir önbilgi olmaksızın, kovana etrafında kendiliğinden yiyecek (çözüm) aramaya başlayan arılardır. İşçi arılar, yiyecek kaynağına ulaşmış ve kaynaktan kovana dönerek yiyecek kaynağının kalitesi, kovana olan uzaklık, yön bilgisi gibi bilgileri diğer arılarla paylaşan arılardır. Gözcü arılar ise işçi arılardan edindiği bilgiye göre hareket ederek yiyecek kaynaklarına ulaşan arılardır.

Arı algoritmasının temel adımları aşağıdaki şekildedir.

- Başlangıç arı popülasyonunu rassal olarak oluştur
- Popülasyonunun uygunluğunu değerlendir
- Tekrarla
  - Komşuluk araması yapılacak bölgeleri belirle
  - Arıları belirlenen bölgelere gönder ve uygunluklarını değerlendir
  - Her bir bölgedeki en iyi uygunluk değerine sahip arıyı seç
  - Kalan arıları rastgele arama için ata ve uygunluklarını değerlendir
- Durdurma kriteri sağlanıncaya kadar

### 3.1 Çözümlerin gösterimi

Çözüm dizisi üç bölümlü olarak tasarlanmıştır. İlk bölüm hücre sayısını (C), ikinci bölüm makine, işçi ve parçaların atandığı hücreleri gösterirken üçüncü bölüm, işçilerin atandığı makineleri ve parçaları göstermektedir. İlk ve ikinci bölümün uzunluğu sabit iken üçüncü bölümün uzunluğu değişkendir. Çözüm dizisinin elemanları tamsayı değer almaktadır.

İlk bölüm ilgili çözümün kaç hücreli bir çözüme ait olduğunu göstermektedir ve tek bir elemana sahiptir. Tasarlanan yapıda her bir çözüm için hücre sayısı  $C$ ,  $[2 - \text{enk}\{P, M, W\}]$  aralığında tamsayı değer almaktadır. İkinci bölüm, makine, işçi ve parçayı temsil eden üç alt bölümden oluşur. Bu bölümün uzunluğu makine, parça ve işçi sayısının toplamı kadardır. İkinci bölümde  $[1 - C]$  aralığında tamsayı değerler yer almakta ve makine, parça ve işçinin hangi hücreye atandığı bilgisini içermektedir. Örneğin Şekil 3'te verilen iki hücreli ( $C = 2$ ) çözümde makine  $\{2, 3\}$ , işçi  $\{3, 4\}$  ve parça  $\{1, 4\}$  hücre 1'e, makine  $\{1, 4\}$ , işçi  $\{1, 2\}$  ve parça  $\{2, 3\}$  hücre 2'ye atanmıştır.

Hücre	Makine				İşçi				Parça			
2	2	1	1	2	2	2	1	1	1	2	2	1
C	M1	M2	M3	M4	W1	W2	W3	W4	P1	P2	P3	P4

Şekil 3. Çözüm dizisinin ilk ve ikinci bölümü için bir örnek.

Figure 3. An example for the first and second part of a solution vector

Çözüm dizisinin üçüncü bölümü, makineleri kullanma becerilerini dikkate alarak işçilerin makine ve parçalara atanmasını simgeler. Üçüncü bölüm, öncelikle parça sayısına bölünmüştür. Her bir parçaya karşı gelen bu bölümler de bu parçalar için gerekli makine sayısına bölünmüştür. Örneğin, 1 No.lu parçanın 3 farklı makinede işlem görmesi gerekiyorsa; çözümün üçüncü bölümündeki bu parça ile ilgili kısım kendi içinde tekrar 3'e bölünür. Her parçanın işlem görmesi gereken makine sayısı farklı olduğu için bu bölümün uzunluğu sabit değildir. Üçüncü bölümdeki tamsayı değerler, işçiyi temsil etmektedir. Böylece hangi işçinin hangi parçaları işleyeceği bilgisi elde edilmektedir. Önerilen algoritmada çözüm dizisinin gelişmiş yapısı, literatürdeki çalışmalardan farklı olarak, ilgili makinedeki parçayı işleyebilecek uygun bir işçi grubundan seçim yapmayı mümkün kılar. Böylece her zaman olurlu çözüm türetilmesi garanti edilmektedir.

Tablo 1'de verilen parça-makine, makine-işçi ve parça-işçi matrisleri dikkate alınarak hazırlanan uygun parça-makine-işçi atamalarını gösteren aday matrisi Tablo 2'de gösterilmiştir.

Tablo 2'deki aday matrisi için bir çözüm dizisinin üçüncü bölümüne örnek Şekil 4'te verilmiştir.

Ele alınan örnekte 4 parça olduğu için üçüncü bölüm her biri bir parçayı gösterecek şekilde dörde bölünmüştür. Her bir parça

için alt bölümler değişken olmaktadır. Örneğin Parça 2, Makine 1 ve Makine 3'de işlem göreceğinden Parça 2'nin temsil edildiği bölüm iki alt bölüme bölünmüştür. Böylece çözüm dizisinin üçüncü bölümündeki dördüncü değer olan 2, İşçi 2'nin Parça 2'yi Makine 1'de işleyeceğini ifade etmektedir. İlgili çözüm dizisi örneğinin tamamı Şekil 5'te verilmiştir.

### 3.2 Komşuluk yapısı

Bir çözümün komşusu türetilirken olurlu çözümler türetebilmek için ilk bölümün değiştirilmesine hiç bir zaman izin verilmemiştir. Böylece ele alınan çözümle aynı hücre sayısına sahip komşu çözümler türetilmiştir. Algoritma, ikinci ve üçüncü bölümün her ikisinde de değiştirilecek eleman sayısının farklı olmasına izin verecek şekilde esnek olarak tasarlanmıştır. İlgili bölümde değiştirilecek elemanların seçimi rassal olarak yapılmaktadır. Seçilen eleman için olurlu tüm atamalar türetilip amaç fonksiyonu değeri en iyi olan, komşu çözüm olarak benimsenmektedir. Şekil 6'da örnek çözüm dizisinin ikinci ve üçüncü bölümünden rasgele seçilen elemanlar için Tablo 1'de verilen matris değerlerine uygun olurlu atamalar gösterilmiştir. İkinci bölümden 1, üçüncü bölümden de 1 adet elemanın değişmesine izin verilen örnekte rassal olarak ikinci bölümden 6. eleman, üçüncü bölümden 7. eleman seçilmiştir. Çözüm iki hücreli olduğu için işçi 2 ya hücre 1'e ya da hücre 2'ye atanabilir. İşçi 2, eldeki çözümde hücre 2'ye atandığı için komşu çözümde hücre 1'e atanacaktır. Üçüncü bölümdeki seçilen eleman ise parça 3'ün makine 2'de hangi işçi ile üretileceği bilgisini içermektedir. Bunun için olurlu olan işçi 2, işçi 3 ve işçi 4'tür. Eldeki çözümde işçi 2 atandığına göre işçi 3 ve işçi 4 bu çözümün komşularıdır. Bu durumda komşu çözümler, Şekil 7'de verildiği gibi olacaktır.

### 3.3 Çözümlerin değerlendirilmesi

İkinci bölümde verilen matematiksel modelin amaç fonksiyonu, arı algoritmasıyla elde edilen çözümlerin başarısını ölçmekte kullanılmıştır. Modelin kısıtlarından hücreye atanabilecek parça, makine ve işçi sayısının alt sınırını ifade edenler (4, 7 ve 8 numaralı kısıtlar) dışındaki kısıtlar, çözümlerin gösterim şekli ve komşuluk yapıları tasarlanırken dikkate alınmış ve böylece olurlu çözümler türetmek mümkün olmuştur.

Çözümlerin gösterimi ve komşuluk yapıları ile sağlanamayan hücre büyüklüğü kısıtları (4, 7 ve 8 numaralı kısıtlar) ise belirli bir ceza katsayısı ile amaç fonksiyonu değerine eklenmiştir. Çözümlerin değeri, aşağıdaki şekilde hesaplanmaktadır:

Çözüm değeri = toplam hücre dışı eleman sayısı + toplam boşluk sayısı + toplam ceza değeri

Toplam ceza değeri =  $E1 * (\text{makine alt sınırından sapma}) + E2 * (\text{işçi alt sınırından sapma}) + E3 * (\text{parça alt sınırından sapma})$

E1, E2 ve E3 ceza katsayılarını ifade etmektedir ve problem boyutuna göre farklı olarak belirlenebilir.

Tablo 1. Parça-makine, makine-işçi ve parça-işçi matrisleri.

Table 1. Part-machine, machine-worker and part-worker incidence matrices.

Parça	Makine				İşçi				Parça	İşçi			
	1	2	3	4	Makine	1	2	3		4	1	2	3
1	0	1	1	1	1	1	1	0	1	1	1	1	1
2	1	0	1	0	2	1	1	1	1	2	1	1	1
3	1	1	1	1	3	1	1	1	1	3	0	1	1
4	1	0	1	0	4	0	1	0	0	4	1	0	1





### 3.4 Kübik hücre oluşturma problemi için arı algoritması

Geliştirilen arı algoritmasının çözümlerin gösterimi, komşuluk yapısı ve çözümlerin değerlendirilmesi aşamalarında kübik hücre oluşturma probleminin özellikleri dikkate alınarak tasarım yapılmıştır. Algoritmanın genel adımlarında ise Tapkan ve diğ.'nin [15] genelleştirilmiş atama probleminin çözümü için önerdikleri arı algoritmasına benzer bir algoritma kullanılmıştır.

Arı algoritmasının adımları aşağıdaki şekildedir.

$m$  : Kâşif arı sayısı

$p$  : İşçi arı sayısı

$e$  : En iyi işçi arı sayısı

$n_{ep}$  :  $e$  adet işçi arının her birine gönderilecek gözcü arı sayısı

$n_{sp}$  :  $p - e$  adet işçi arının her birine gönderilecek gözcü arı sayısı ( $n_{sp} < n_{ep}$ )

$maksIter$  : Enbüyük iterasyon sayısı, durdurma kriteri

$maksLimit$  : Bir çözüm için iyileşme olmaksızın izin verilen en fazla iterasyon sayısı

$limitsayaci(\sigma^p)$  :  $\sigma^p$  çözümü için iyileşme olmaksızın geçen iterasyon sayısı

$\sigma^p$  :  $p$ . işçi arının çözümü

$fit(\sigma^p)$  :  $\sigma^p$  çözümünün değeri

**Algoritma**

Gerekli verileri topla, parametreleri belirle

$m$  adet kâşif arı için çözümleri rassal olarak oluştur.

Çözümleri değerlendir

$I = 0$

**Tekrarla**

Eniyi çözüm değerine sahip  $p$  tane çözümü işçi arı olarak belirle.

İşçi arıların eniyi  $e$  tanesinin her birine  $n_{ep}$  tane gözcü arı ata.

Kalan  $p - e$  tane işçi arının her birine  $n_{sp}$  tane gözcü arı ata.

$t = 0$

**Tekrarla**

$k = 0$

**Tekrarla**

$e$  tane işçi arıya atanmış her gözcü arının komşularını ara.

Eğer  $fit(\sigma^{komsucozum}) < fit(\sigma^p)$  ise  $\sigma^p = \sigma^{komsucozum}$  ve

$limitsayaci(\sigma^p) = 0$ , dd.  
 $limitsayaci(\sigma^p) = limitsayaci(\sigma^p) + 1$

$k = k + 1$

**Sağlandığı sürece** ( $k < n_{ep}$ )

$k = 0$

**Tekrarla**

$p - e$  tane işçi arıya atanmış her gözcü arının komşularını ara.

Eğer  $fit(\sigma^{komsucozum}) < fit(\sigma^p)$  ise  $\sigma^p = \sigma^{komsucozum}$  ve

$limitsayaci(\sigma^p) = 0$ , dd.

$limitsayaci(\sigma^p) = limitsayaci(\sigma^p) + 1$

$k = k + 1$

**Sağlandığı sürece** ( $k < n_{sp}$ )

Eğer ( $limitsayaci(\sigma^p) > maksLimit$ ) ise yeni bir kâşif arı çözümü oluştur.

$t = t + 1$

**Sağlandığı sürece** ( $t < p$ )

En iyi çözümü güncelle

Rassal olarak ( $m - p$ ) tane yeni kâşif arı çözümleri oluştur.

$I = I + 1$

**Sağlandığı sürece** ( $I < maksIter$ )

Doğadaki arılarda yiyecek arama işlemini önce kâşif arılar başlatır ve rastgele yiyecek kaynakları seçerler. Ayrıca yiyecek toplama süreci boyunca popülasyonun belli bir yüzdesi kâşif arılar olarak kalır. Benzer olarak arı algoritması da ( $m$ ) tane kâşif arının araştırma uzayına rastgele yerleştirilmesi ile başlar. Daha sonra kâşif arılarca ziyaret edilen noktaların uygunlukları değerlendirilir. Doğadaki arılar incelendiğinde, bir arı kolonisinin yeterli yiyecek kaynağını sağlayabilmek için oldukça geniş alanı aramaları gerekebilmektedir. Prensipite nektar miktarı fazla olan ve kolayca tüketilebilen yiyecek kaynakları daha fazla arı tarafından, az miktarda nektara sahip yiyecek kaynakları ise daha az arı tarafından ziyaret edilecektir. Bu nedenle en iyi uygunluk değerine sahip arılar ( $p$ ) komşuluk araması için seçilir. Seçilen arıların komşuluğunda araştırma yapılır ve daha umut verici çözümleri temsil eden en iyi  $e$  bölgeye, seçilen diğer bölgelere ( $p - e$ ) göre daha fazla arı gönderilerek daha detaylı arama yapılır. Yeni popülasyona, her bölgedeki en iyi uygunluk değerine sahip arılar seçilir. Eğer herhangi bir çözüm  $maksLimit$  adet iterasyon boyunca iyileştirilememişse bu çözüm terkedilir ve rassal olarak yeni bir kâşif arı çözümü oluşturulur. Popülasyondaki diğer ( $m - p$ ) tane kâşif arı, yeni potansiyel çözümler elde etmek için rastgele olarak araştırma uzayına atanırlar. Böylece her yeni popülasyon, seçilen her bölgenin temsilcileri ve rastgele arama yapan kâşif arılar olmak üzere iki parçadan oluşacaktır. Algoritmanın adımları önceden belirlenmiş iterasyon sayısı  $maksIter$  kadar tekrarlanır.

## 4 Deneysel sonuçlar

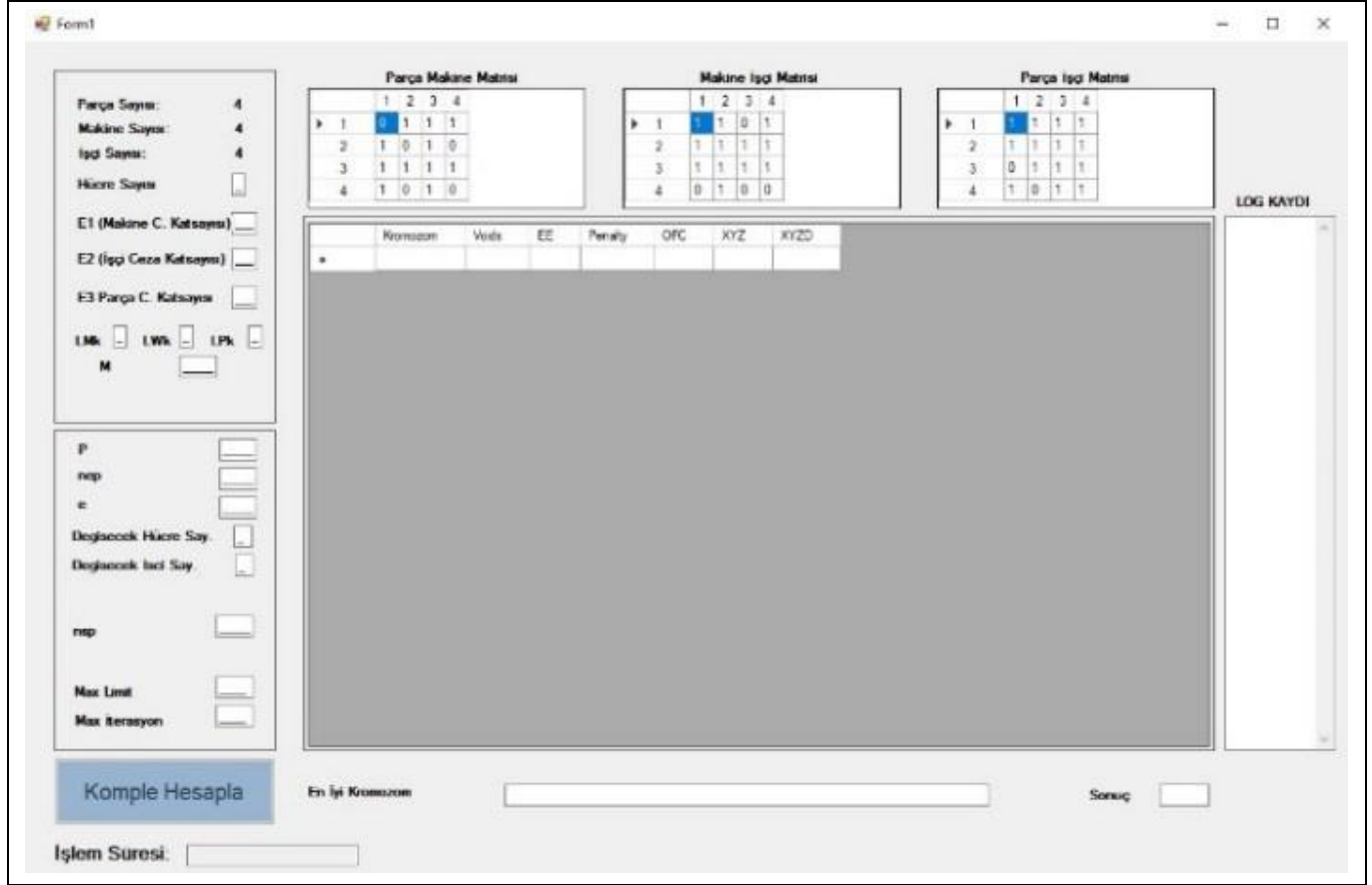
Geliştirilen arı algoritması, C# dilinde kodlanmış ve literatürden alınan test problemleri ile 2.20 GHz CPU, 8.00 GB RAM özelliklere sahip Intel Core i7 işlemciye sahip PC kullanılarak test edilmiştir. Hazırlanan programın ekran görüntüsü Şekil 8'de verilmiştir.

Algoritmanın performansı Mahdavi ve diğ.'den [5] alınan Veri Seti-A ve Buruk Şahin ve Alpay'dan [9] alınan Veri Seti-B kullanılarak test edilmiştir. Veri Seti-A'daki problemlerin parça, makine ve işçi sayıları Tablo 3'te, Veri Seti-B'dekilerinki ise Tablo 4'te verilmiştir.

Tablolarda ayrıca her bir problemin alındığı kaynakta hangi hücre sayısı için çözüm arandığı bilgisi de verilmiştir. Veri Seti-B'de yer alan P8 problemi P9 ile, P10 problemi de P11 ile aynı olmasına rağmen farklı hücre sayıları için çözüm arandığından

farklı bir problem olarak sunulmuştur.  $LM_k$ ,  $LW_k$  ve  $LP_k$  parametrelerinin değerleri diğer çalışmalarda olduğu gibi 1 olarak kabul edilmiştir. Makine, işçi ve parça ceza katsayıları da (E1, E2, E3) tüm problemlerde 100 olarak alınmıştır.

Algoritma parametreleri, genel öneriler ve deneyimlere göre belirlenmiştir (Tablo 5).



Şekil 8. Geliştirilen arı algoritmasının ekran görüntüsü.

Figure 8. Screen shot of the developed bees algorithm.

Tablo 3. Veri seti-A.

Table 3. Data set-A.

Problem	Parça sayısı	Makine sayısı	İşçi sayısı	Hücre sayısı
P1	4	4	4	2
P2	5	4	5	2
P3	6	5	5	2
P4	10	7	4	2
P5	10	7	6	3

Tablo 4. Veri seti-B.

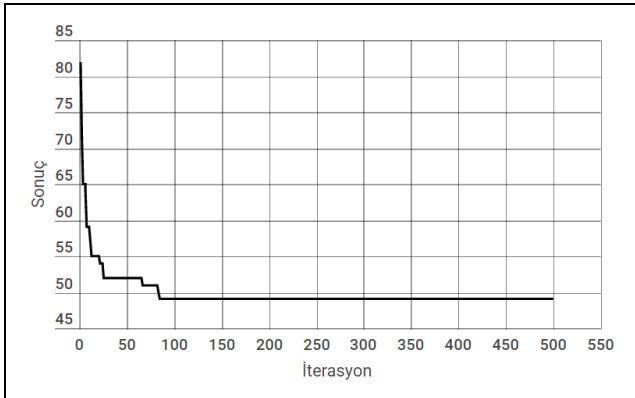
Table 4. Data set-B.

Problem	Parça sayısı	Makine sayısı	İşçi sayısı	Hücre sayısı
P6	12	8	6	3
P7	12	8	7	3
P8	15	10	6	3
P9	15	10	6	4
P10	20	10	6	3
P11	20	10	6	4

Tablo 5. Arı algoritması parametreleri.  
Table 5. Parameters of the bees algorithm.

Parametre	Veri Seti-A	Veri Seti-B
$m$	100	200
$p$	30	50
$e$	10	20
$n_{ep}$	30	50
$n_{sp}$	20	30
Makslimit	100	250
Maksiter	500	1000

Örnek olarak seçilen P8 probleminin önerilen algoritma ile çözümünde amaç fonksiyonu değerinin iterasyonlar boyunca değişimi, Şekil 9'daki yakınsama grafiğinden görülebilir. Grafikte yatay eksen iterasyon sayısını, dikey eksen de amaç fonksiyonu değerini göstermektedir. Amaç fonksiyonu değerinin iterasyonlar boyunca kararlı bir şekilde iyileştiği dikkati çekmektedir. Grafikte dikkati çeken nokta ise, algoritmanın 500 iterasyon yapmasına izin verilmesine rağmen aslında 86. iterasyondan itibaren en iyi çözümde herhangi bir iyileşme görülmemiştir. Bir çözüm yolunun algoritma niteliği kazanabilmesi için mutlaka sonlu sayıda adım ile sonuca ulaşması beklenmektedir. Bu grafikte de, en büyük değer eğrisinin, yatay eksene paralel bir konum alması ve bu konumu makul bir süre devam ettirmesi, artık daha fazla iyileşme beklenmeyeceği kanısını uyandırıp, algoritmanın durdurulacağı noktayı belirlemektedir.



Şekil 9. P8 problemi için yakınsama grafiği.

Figure 9. Convergence graph for problem P8.

#### 4.1 Hücre sayısının bilindiği durum için arı algoritması sonuçları

Önerilen algoritma öncelikle literatürde ele alındığı şekilde hücre sayısının önceden bilindiği ve amaç fonksiyonunun toplam hücre dışı eleman ve boşluk sayılarının enküçüklenmesi olduğu durum için sınanmıştır. Her bir test problemi, önerilen arı algoritması ile 10 kez çözülmüştür. Hücre sayıları karşılaştırma yapılan yöntemlerle aynı alınmıştır. Elde edilen sonuçlar Mahdavi ve diğ. [5] ve Buruk Şahin ve Alpay [9] tarafından elde edilen sonuçlarla karşılaştırılmıştır (Tablo 6 ve Tablo 7). Tablolarda her bir problem için önerilen arı algoritması ile elde edilen çözümün boşluk sayısı (B), hücre dışı eleman sayısı (HDE), 10 tekrarda elde edilen toplam (boşluk ve hücre dışı eleman sayılarının toplamı) değerinin eniyisi, ortalaması ve standart sapması ve saniye cinsinden işlem süresi verilmiştir. Tablolardan görüldüğü üzere, çözülen 11 test probleminden 9 tanesinde bilinen eniyi çözümlere

ulaşmış, 2 tanesinde ise literatürdeki çözümler iyileştirilmiştir. Önerilen arı algoritmasının çözüm süresi açısından avantajlı olduğu da görülmektedir.

#### 4.2 Hücre sayısının bilinmediği durum için arı algoritması sonuçları

Önerilen algoritmanın başarılı olduğu literatürden alınan test problemleri kullanılarak görüldükten sonra algoritma, farklı amaç fonksiyonu ve durumları dikkate alacak şekilde genişletilerek test edilmiştir. Hücre sayısının bilindiği ve bilinmediği durumda test problemlerinin önerilen arı algoritması ile çözümüyle elde edilen sonuçlar Tablo 8'de verilmiştir. Tablodan görülebileceği gibi hücre sayısının belirli olmadığı ve algoritmanın belirlediği durumda daha iyi amaç fonksiyonu değerine sahip çözümler elde edilmiştir. P9 ve P11 problemleri sırasıyla P8 ve P10 problemleri ile aynı verilere sahip olduklarından bu problemler için hücre sayısının bilinmediği durum için sonuçları belirtilmemiştir.

İki boyutlu hücre oluşturma probleminde performans ölçütü olarak genellikle Gruplama Etkinliği (Grouping Efficacy) (GE) ölçütünü kullanılmaktadır. GE aşağıdaki şekilde hesaplanmaktadır.

$$GE = \frac{e - e_0}{e + e_v} \quad (11)$$

Burada,  $e$  değeri matristeki 1'lerin toplam sayısı,  $e_0$  değeri çözümdeki hücre dışı eleman sayısı ve  $e_v$  değeri çözümdeki boşluk sayısıdır.

Küçük hücre oluşturma probleminde GE değerinin hesaplanmasında kullanılacak matristeki toplam 1'lerin sayısının bulunmasında, uygun parça-makine-işçi atamalarını gösteren aday matrisinden faydalanılmaktadır.

P1 için  $e$  değeri Tablo 2'deki uygun atamaların sayısına eşit olup 30 değerine sahiptir ve yapılan çözümler sonrası en iyi GE değerine sahip çözüm için boşluk sayısı 4, hücre dışı eleman sayısı 3 bulunmuştur. GE değeri hesaplandığında  $(30-3)/(30+4)=79.41$  sonucu elde edilmektedir. Test problemleri önerilen arı algoritması ile hücre sayısının bilindiği ve bilinmediği durumlar için amaç fonksiyonu gruplama etkinliğinin enbüyüklenmesi olarak çözülmüştür. Sonuçlar Tablo 9 ve Tablo 10'da verilmiştir. Sonuçlar incelendiğinde amaç fonksiyonu olarak hücre dışı eleman ve boşluk sayılarının toplamının enküçüklenmesini kullanarak daha iyi toplam değerler elde edilirken, gruplama etkinliği amaç fonksiyonu ile daha düşük hücre dışı eleman sayılarına sahip çözümlerin elde edildiği görülmektedir. Bu durum GE ölçütünün hücre dışı eleman sayısının düşük olmasına daha fazla önem vermesinden kaynaklanmaktadır. Hücre dışı taşımaların hücre içi taşımalara oranla daha önemli olduğu durumlarda bu ölçütün kullanımının daha iyi olacağı açıktır.



Tablo 6. Veri seti-A için elde edilen sonuçlar.

Table 6. The results for data set-A.

Mahdavi ve Aalaei ve diğ. [5]			Buruk Şahin ve Alpay [9]			Önerilen Arı Algoritması					
B	HDE	Toplam	Toplam	Süre(s)	B	HDE	Toplam (eniye)	Toplam (ort)	Toplam (ss)	Süre(s)	
P1	4	3	7	7	1	4	3	7	7.00	0	1
P2	10	1	11	11	2	10	1	11	11.00	0	1
P3	14	4	18	18	3	14	4	18	18.00	0	1
P4	20	14	34	34	9	20	14	34	34.00	0	1
P5	19	6	25	25	143	19	6	25	25.20	0.42	6

Tablo 7. Veri seti-B için elde edilen sonuçlar.

Table 7. The results for data set-B.

Buruk Şahin ve Alpay [9]				Önerilen Arı Algoritması					
	Toplam (eniye)	Toplam (ort)	Süre (s)	B	HDE	Toplam (eniye)	Toplam (ort)	Toplam (ss)	Süre (s)
P6	29	29.66	33	11	18	29	29.00	0	6
P7	33	34.33	39	14	19	33	33.40	0.52	8
P8	49	49.33	40	25	24	49	49.10	0.32	12
P9	37	38.66	42	13	22	35	36.20	1.03	14
P10	60	62.00	49	28	32	60	60.90	0.32	20
P11	47	49.66	54	19	26	45	45.90	0.99	26

Tablo 8. Hücre sayısının bilindiği ve bilinmediği durumda elde edilen arı algoritması sonuçlarının karşılaştırılması.

Table 8. Comparison of the bees algorithm results for the cases that the number of cells is known and not known.

Hücre Sayısı Biliniyor								Hücre Sayısı Bilinmiyor						
	C	B	HDE	Toplam (eniye)	Toplam (ort)	Toplam (ss)	Süre(s)	C	B	HDE	Toplam (eniye)	Toplam (ort)	Toplam (ss)	Süre(s)
P1	2	4	3	7	7.00	0	1	3	1	5	6	6.00	0	1
P2	2	10	1	11	11.00	0	1	3	3	1	4	4.00	0	1
P3	2	14	4	18	18.00	0	1	4	1	4	5	5.00	0	1
P4	2	20	14	34	34.00	0	1	4	2	17	19	19.50	0.53	5
P5	3	19	6	25	25.20	0.42	6	5	3	5	8	10.20	2.10	7
P6	3	11	18	29	29.00	0	6	6	3	14	17	18.00	0.67	14
P7	3	14	19	33	33.40	0.52	8	6	5	6	11	13.30	1.57	17
P8	3	25	24	49	49.10	0.32	12	6	8	17	25	27.30	1.42	23
P9	4	13	22	35	36.20	1.03	14	-	-	-	-	-	-	-
P10	3	28	32	60	60.90	0.32	20	6	10	23	33	34.80	1.48	28
P11	4	19	26	45	45.90	0.99	26	-	-	-	-	-	-	-

Tablo 9. Hücre sayısının bilindiği durum için gruplama etkinliği değerleri.

Table 9. Grouping efficacy values for the case that the number of cells is known.

B+HDE							GE						
	C	B	HDE	Toplam (eniye)	Toplam (ort)	Süre (s)	B	HDE	Toplam (eniye)	GE	Toplam (ort)	Toplam (ss)	Süre (s)
P1	2	4	3	7	7.00	1	4	3	7	79.41	79.41	0	1
P2	2	10	1	11	11.00	1	10	1	11	50.00	50.00	0	1

Tablo 9. Devami.

Table 9. Continued.

	B+HDE						GE						
	C	B	HDE	Toplam (eniye)	Toplam (ort)	Süre(s)	B	HDE	Toplam (eniye)	GE	Toplam (ort)	Toplam (ss)	Süre (s)
P3	2	14	4	18	18.00	1	14	4	18	43.75	43.75	0	1
P4	2	20	14	34	34.00	1	26	11	37	45.59	45.59	0	1
P5	3	19	6	25	25.20	6	30	1	31	39.22	39.22	0	2
P6	3	11	18	29	29.00	6	23	10	33	37.74	37.74	0	7
P7	3	14	19	33	33.40	8	28	6	34	46.03	46.03	0	8
P8	3	25	24	49	49.10	12	38	15	53	33.75	33.75	0	8
P9	4	13	22	35	36.20	14	22	15	37	42.19	42.19	0	9
P10	3	28	32	60	60.90	20	42	20	62	35.42	35.42	0	9
P11	4	19	26	45	45.90	26	27	19	46	43.21	42.84	0.26	11

Tablo 10. Hücre sayısının bilinmediği durum için gruplama etkinliği değerleri.

Table 10. Grouping efficacy values for the case that the number of cells is not known.

	B+HDE						GE							
	C	B	HDE	Toplam (eniye)	Toplam (ort)	Süre(s)	C	B	HDE	Toplam (eniye)	GE	Toplam (ort)	Toplam (ss)	Süre(s)
P1	3	1	5	6	6.00	1	3	1	5	6	80.65	80.65	0	1
P2	3	3	1	4	4.00	1	3	3	1	4	73.33	73.33	0	1
P3	4	1	4	5	5.00	1	4	1	4	5	73.68	70.93	2.90	3
P4	4	2	17	19	19.50	5	3	7	13	20	59.18	59.18	0	4
P5	5	3	5	8	10.20	7	4	8	5	13	55.17	45.15	6.47	10
P6	6	3	14	17	18.00	14	5	7	12	19	48.65	46.54	1.23	29
P7	6	5	6	11	13.30	17	6	4	8	12	69.23	66.04	2.43	35
P8	6	8	17	25	27.30	23	4	22	15	37	42.19	39.25	2.01	31
P10	6	10	23	33	34.80	28	5	19	20	39	46.58	42.87	1.87	38

## 5 Sonuçlar

Hücre oluşturma alanında yapılan çalışmaların büyük çoğunluğu problemin iki boyutu olan parça ve makineye odaklanmaktadır. Diğer taraftan son yıllarda, makinelerde iş yapma konusundaki önemli rolleri nedeniyle işçiler üçüncü boyut olarak vurgulanmaya başlanmıştır. Üretkenliğin artırılmasının bir yolu uygun işçiye uygun makinelerle uygun parçaları yaptırmaktır. İşçiler, makinelerde iş yapmakta önemli rol oynamakta olduğundan, çalışanların hücrelere atanması hücresel üretimin tam olarak kullanılması için çok önemli bir faktör haline gelmektedir. Bu çalışmada, hücre sayısının belirli olmadığı durumda üç boyutlu parça-makine-işçi temas matrisini dikkate alan kübik hücre oluşturma probleminin çözümünde arı algoritması ilk kez kullanılmıştır. Geliştirilen algoritmanın performansı, literatürden alınan problemler kullanılarak test edilmiştir. Önerilen arı algoritmasının eniyi hücre sayısını da belirlemesine olanak sağlanmıştır. Böylece hücre sayısının belirli olduğu duruma göre daha iyi çözümler elde edilmiştir. Son olarak iki boyutlu hücre oluşturma problemlerinde sıklıkla kullanılan bir performans ölçütü olan gruplama etkinliği, üç boyutlu probleme uyarlanmış ve geliştirilen algoritma, gruplama etkinliği amaç fonksiyonu ile de test edilmiştir. Bu durumda da hücre dışı eleman sayısı açısından daha iyi çözümlere ulaşılabildiği görülmüştür.

Bu çalışma arı algoritmasının kübik hücre oluşturma problemi çözümünde ilk kez kullanılması nedeniyle algoritmanın performansının değerlendirilmesi açısından önemlidir. Bu çalışmanın kapsamı dışında kalan ancak ileride yapılacak çalışmalarda ele alınabilecek çalışma konularından ilki, hücre oluşumu sırasında kapasite kısıtını da ele almanın daha gerçekçi sonuçlar elde edilmesine olanak sağlamasıdır. Bu nedenle algoritmanın bu kısıtı da göz önünde bulunduracak şekilde geliştirilmesi faydalı olacaktır. Ayrıca problemin çok amaçlı yapısı nedeniyle kübik hücre oluşturma probleminin çözümünde çok amaçlı yöntemlerin kullanılmasının da faydalı olacağı düşünülmektedir.

## 6 Conclusions

The vast majority of work done in this area is focused on the part and machine, which are the two basic dimensions of the cell formation problem. On the other hand, in recent years, workers have been emphasized as a third dimension because of their important role in their work on machines. The way to increase the productivity is to attain suitable parts and machines for the appropriate workmanship. Since workers play an important role in doing work on machines, the assignment of workers to cells is a very important factor to make use of cellular production. In this study, the bee algorithm was used

for the first time in solving the cubic cell formation problem that considers the three-dimensional part-machine-worker incidence matrix when the number of cells is not certain. The performance of the developed algorithm has been tested by using problems from the literature. The proposed bee algorithm was also allowed to determine the optimal cell number. Thus, better solutions were obtained depending on certain number of cells. Finally, the grouping efficacy, which is a performance measure commonly used in two-dimensional cell formation problems, was adapted to the three-dimensional problem and the developed algorithm was tested with the grouping efficacy objective function. In this case, it has been observed that better solutions can be obtained in the number of exceptional elements.

This study is important in terms of evaluating the performance of the algorithm since the bee algorithm is used for the first time in the solution of cubic cell formation problem. The first of the study subjects that are out of the scope of this study but may be discussed in future studies is that it allows for more realistic results to address capacity constraints during cell formation. In addition, due to the multi-objective nature of the problem, it is thought that using multi-objective methods will be useful in solving the problem of cubic cell formation.

## 7 Kaynaklar

- [1] Bidanda B, Ariyawongrat P, Needy KL, Norman BA, Tharmmaphornphilas W. "Human related issues in manufacturing cell design, implementation, and operation: a review and survey". *Computers & Industrial Engineering*, 48(3), 507-523, 2005.
- [2] Min H, Shin D. "Simultaneous formation of machine and human cells in group technology: A multiple objective approach". *International Journal of Production Research*, 31(10), 2307-2318, 1993.
- [3] Li ML. "The algorithm for integrating all incidence matrices in multi-dimensional group technology". *International Journal of Production Economics*, 86(2), 121-131, 2003.
- [4] Mahdavi I, Aalaei A, Paydar MM, Solimanpur M. "Designing a mathematical model for dynamic cellular manufacturing systems considering production planning and worker assignment". *Computers & Mathematics with Applications*, 60(4), 1014-1025, 2010.
- [5] Mahdavi I, Aalaei A, Paydar MM, Solimanpur M. "A new mathematical model for integrating all incidence matrices in multi-dimensional cellular manufacturing system". *Journal of Manufacturing Systems*, 31(2), 214-223, 2012.
- [6] Nikoofarid E, Aalaei A. "Production planning and worker assignment in a dynamic virtual cellular manufacturing system". *International Journal of Management Science and Engineering Management*, 7(2), 89-95, 2012.
- [7] Aalaei A, Shavazipour B. "The tchebycheff norm for ranking DMUs in cellular manufacturing systems with assignment worker". *International Journal of Applied Operational Research*, 3(3), 41-57, 2013.
- [8] Bootaki B, Mahdavi I, Paydar MM. "A hybrid GA-AUGMECON method to solve a cubic cell formation problem considering different worker skills". *Computers & Industrial Engineering*, 75, 31-40, 2014.
- [9] Buruk Şahin Y, Alpay Ş. "A metaheuristic approach for a cubic cell formation problem". *Expert Systems with Applications*, 65, 40-51, 2016.
- [10] Heragu SS, Chen J-S. "Optimal solution of cellular manufacturing system design: Benders' decomposition approach". *European Journal of Operational Research*, 107(1), 175-192, 1998.
- [11] Gharbi I, Mukattash A. "Efficiency vs. number of cells in cell formation". *2018 Third International Conference on Electrical and Biomedical Engineering, Clean Energy and Green Computing (EBECEGC)*, Beirut-Lebanon, 25-27 April 2018.
- [12] Utkina I, Batsyn M, Batsyna E. "A branch and bound algorithm for a fractional 0-1 programming problem". In: Kochetov Y, Khachay M, Beresnev V, Nurminski E, Pardalos P. (Eds): *Discrete Optimization and Operations Research. DOOR 2016. Lecture Notes in Computer Science*, vol 9869, Springer, 244-255, 2016.
- [13] Bychkov I, Batsyn M. "An efficient exact model for the cell formation problem with a variable number of production cells". *Computers & Operations Research*, 91, 112-120, 2018.
- [14] Utkina IE, Batsyn MV, Batsyna EK. "A branch-and-bound algorithm for the cell formation problem". *International Journal of Production Research*, 56(9), 3262-3273, 2018.
- [15] Tapkan P, Özbakir L, Baykasoğlu A. "Arı algoritması ve genelleştirilmiş atama problemi: farklı komşuluk yapılarının karşılaştırılması". *Endüstri Mühendisliği Dergisi*, 21(2), 2-13, 2010.
- [16] Özbakir L, Baykasoğlu A, Tapkan P. "Bees algorithm for generalized assignment problem". *Applied Mathematics and Computation*, 215, 3782-3795, 2010.
- [17] Tapkan P, Özbakir L, Baykasoğlu A. "Bee algorithms for parallel two-sided assembly line balancing problem with walking times". *Applied Soft Computing*, 39, 275-291, 2016.
- [18] Tapkan P, Özbakir L, Baykasoğlu A. "Bees Algorithm for constrained fuzzy multi-objective two-sided assembly line balancing problem". *Optimization Letters*, 6, 1039-1049, 2012.
- [19] Pham DT, Afify A, Koç E. "Manufacturing Cell Formation Using the Bees Algorithm". *3rd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2007)*, Cardiff, UK, 2-13 July 2007.
- [20] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M. "The bees algorithm-A novel tool for complex optimisation problems". *Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006)*, Cardiff, UK, 3-14 July 2006.