Endüstri Mühendisliği Dergisi
Cilt: 21 Sayı: 2 Sayfa: (2-13)

ARI ALGORİTMASI VE GENELLEŞTİRİLMİŞ ATAMA PROBLEMİ: FARKLI KOMŞULUK YAPILARININ KARŞILAŞTIRILMASI

Pınar TAPKAN[1], Lale ÖZBAKIR[1], Adil BAYKASOĞLU[2]*

[1]Erciyes Üniversitesi, Endüstri Mühendisliği Bölümü, Kayseri
[2]Gaziantep Üniversitesi, Endüstri Mühendisliği Bölümü, Gaziantep

pinartan@erciyes.edu.tr, lozbakir@erciyes.edu.tr, baykasoglu@gantep.edu.tr

Geliş Tarihi: 15 Haziran 2008; Kabul Ediliş Tarihi: 19 Mart 2010
Bu makale 2 kez düzeltilmek üzere 86 gün yazarlarda kalmıştır.

ÖZET

Arı Algoritması popülasyon tabanlı yeni bir arama algoritması olup sürü zekâsına dayalı metasezgisel yöntemlerden birisidir. Algoritma gerçek bal arılarının yiyecek arama davranışlarını modellemeye dayanmakta olup bilimsel yazında kombinatoryel ve genellikle de sürekli optimizasyon problemlerinin çözümünde kullanılmıştır. Diğer taraftan Genelleştirilmiş Atama Problemi NP-zor bir problem olup kombinatoryel tamsayılı optimizasyon problemi olarak formüle edilebilmektedir. Bu çalışmada öncelikle Arı Algoritması, Genelleştirilmiş Atama Problemi'nin çözümü için geliştirilmiş ve kaydırma, değiştirme, çift kaydırma ve çıkarım zinciri komşuluk yapılarının Arı Algoritması'nın performansı üzerindeki etkileri incelenmiştir.

Anahtar Kelimeler: Sürü optimizasyonu, arı algoritması, genelleştirilmiş atama problemi

BEES ALGORITHM AND GENERALIZED ASSIGNMENT PROBLEM: COMPARISON OF DIFFERENT NEIGHBORHOOD STRUCTURES

ABSTRACT

Bees Algorithm is a population based new search algorithm which is one of the meta heuristic techniques based on swarm intelligence. Bees Algorithm depends on to model natural behavior of real honey bees in food foraging and is used to obtain solutions for combinatorial and generally continuous optimization problems in the literature. On the other hand, Generalized Assignment Problem is known as an NP-Hard problem and can be formulated as a combinatorial integer optimization problem. In this study, firstly Bees Algorithm is modified to solve Generalized Assignment Problem and the effects of shift, swap, double shift, and ejection chain neighborhood structures on the performance of Bees Algorithm is analyzed.

Keywords: Swarm optimization, bees algorithm, generalized assignment problem

* İletişim yazarı

1. GİRİŞ

Günümüzde karmaşık optimizasyon problemlerinin modellenmesi ve çözülmesinde doğal benzetimlerin kullanımına yönelik bir eğilim vardır. Bunun sebebi klasik optimizasyon algoritmalarının büyük boyutlu kombinatoryel, tamsayılı ve doğrusal olmayan matematiksel programların çözümünde yeterli olmamasıdır. Klasik optimizasyon algoritmalarının çözüm stratejileri genellikle amaç fonksiyonu ve kısıtların türüne (doğrusal, doğrusal olmayan vb.) ve modellemede kullanılan değişkenlerin türüne (tamsayı, gerçek sayı vb.) bağlıdır. Bahsedilen algoritmaların verimliliği ise çözüm alanının boyutuna, modellemede kullanılan değişken ve kısıt sayısına ve çözüm alanının yapısına (konveks, konkav vb.) bağlıdır. Diğer taraftan klasik optimizasyon algoritmaları farklı tipte değişkenler, amaç fonksiyonu ve kısıtlar içeren problem formülasyonlarına uygulanabilecek genel bir çözüm stratejisi sunmamaktadır (Baykasoğlu, 2006).

Ancak birçok optimizasyon problemi aynı formülasyon içinde farklı tiplerde değişkenler, amaç fonksiyonları ve kısıtlar içermektedir. Dolayısıyla klasik optimizasyon teknikleri bu tür problemlerin çözümü için yeterli olmamaktadır. Araştırmacılar çeşitli optimizasyon problemlerini klasik optimizasyon yöntemlerine uyarlamak için oldukça çaba göstermişlerdir. Ancak bir gerçek hayat problemini belli bir çözüm yöntemine uyacak şekilde modellemek genellikle pek kolay değildir. İşte klasik optimizasyon tekniklerinin bu yetersizliklerini aşabilmek için problemden ve modelden bağımsız olan, doğadan esinlenmiş sezgisel optimizasyon algoritmaları önerilmektedir. Bu teknikler hem etkin hem de daha esnek olup belirli problem gereksinimlerine göre uyarlanabilmektedir (Baykasoğlu, 2001).

Doğadan esinlenen algoritmaların yeni bir dalı olan sürü zekâsı yaklaşımı, böceklerin içgüdüsel problem çözme becerilerini kullanan etkili metasezgisel yöntemler geliştirebilmek için böcek davranışlarının modellenmesine odaklanmıştır. Böcekler arasındaki etkileşimin bir sonucu olan kolektif zekânın en önemli parçalarından biri ise bireysel böcekler arasındaki bilgi paylaşımıdır. Bu tür etkileşimli davranışa örnek olarak, bal arılarının buldukları yiyecek kaynağının kalitesi hakkındaki bilgiyi paylaştıkları, salınım dansı verilebilir. Bu dans aracılığıyla kaliteli bir yiyecek kaynağı bulan arılar, yiyecek kaynağı hakkındaki yön, uzaklık ve nektar miktarı bilgilerini diğer arılarla paylaşır. Bu başarılı mekanizma sayesinde koloni, kaliteli yiyecek kaynaklarının olduğu bölgelere yönlendirilebilmektedir. Baykasoğlu vd. (2007) tarafından da ifade edildiği gibi arı kolonisi temelli algoritmalarda temelde üç tip arı sınıfı tanımlanmaktadır:

Kâşif Arılar: Arı sistemi içerisinde tamamen bağımsız davranarak yeni yiyecek kaynaklarını herhangi bir bilgi kullanmadan arayan arı grubudur.

Görevlendirilmiş Arılar: Yiyecek kaynağına ulaşmış ve kaynaktan kovana dönerek yiyecek kaynağına ilişkin bilgileri (yön, uzaklık, yiyecek kaynağının kalitesi vb.) diğer arılarla paylaşmak üzere salınım dansını gerçekleştiren arılardır.

İzci Arılar: Görevli arıların salınım dansını izleyerek yiyecek kaynakları hakkında bilgi edinen ve daha sonra bu bilgiye göre hareket ederek yiyecek kaynaklarına ulaşan arılardır.

Arıların yiyecek arama davranışları, öğrenme, hatırlama ve bilgi paylaşma özellikleri sürü zekâsının en ilgi çekici araştırma alanlarından birisidir. Bal arılarının davranış özellikleri temel alınarak geliştirilmiş algoritmaların bir sınıflandırması ve geniş bir yazın araştırması Baykasoğlu vd. (2007) tarafından yapılmıştır. Bu çalışmanın kapsamı gereği sadece yiyecek arama davranışına dayalı optimizasyon algoritmaları ile ilgili çalışmalara değinilecektir. Lucic ve Teodorovic (2001, 2002, 2003a) ve Lucic (2002) karmaşık ulaştırma problemlerinin çözümünde kolektif arı zekâsı uygulamalarını araştırmışlar ve önerdikleri Arı Sistemi (AS) algoritmasını gezgin satıcı problemine uygulamışlardır. Lucic (2002), Lucic ve Teodorovic (2003b) stokastik araç rotalama problemine iyi çözümler üretebilmek için AS ile bulanık mantık yaklaşımını birleştirmişlerdir. Teodorovic ve Dell'Orco (2005) AS algoritmasını genelleştirerek deterministik ve stokastik kombinatoryel problemleri çözebilen Arı Koloni Optimizasyonu (AKO) algoritmasını önermişlerdir. Wedde vd. (2004)

telekomünikasyon ağında rotalama için arı davranışlarına dayalı bir rotalama protokolü sunmuşlardır. Chong vd. (2006) atölye tipi çizelgeleme problemini çözmek için bal arılarının yiyecek arama davranışını kullanan yeni bir yaklaşım getirmişlerdir. Quijano ve Passino (2007) kaynak paylaştırma problemi için bal arılarının yiyecek arama davranışları üzerine kurulu bir algoritma geliştirmişlerdir. Baştürk ve Karaboğa (2006), Karaboğa ve Baştürk (2007), Yang (2005), Pham vd. (2006a) sürekli optimizasyon problemlerinin çözümü için farklı algoritmalar önermişlerdir.

Bu çalışmada temel alınan Arı Algoritması (AA) ilk olarak Pham vd. (2006a) tarafından önerilmiş olup, bal arılarının yiyecek arama davranışını taklit eden popülasyon tabanlı bir arama algoritmasıdır. AA örüntü tanıma için yapay sinir ağlarının eğitimi (Pham vd., 2006b, 2006c, 2006d, 2006e, 2006b-e, 2007a), üretim hücrelerinin biçimlendirilmesi (Pham vd., 2007b), bir üretim makinesinde işlerin çizelgelenmesi (Pham vd., 2007c), bir tasarım problemine çoklu uygun çözümler bulma (Pham vd., 2007d), veri kümeleme (Pham vd., 2007e), mekanik bileşen tasarımının optimizasyonu (Pham vd., 2007f), çok amaçlı optimizasyon (Pham ve Ghanbarzadeh, 2007) gibi değişik problemlere son yıllarda başarıyla uygulanmıştır. Bu çalışmada ise kaydırma, değiştirme, çift kaydırma ve çıkarım zinciri komşuluk yapılarının AA'nın performansı üzerindeki etkileri incelenmiştir.

2. GENELLEŞTİRİLMİŞ ATAMA PROBLEMİ İÇİN ARI ALGORİTMASI

Bu bölümde AA'nın genel yapısı (Şekil 1) ve Genelleştirilmiş Atama Problemi (GAP) için Geliştirilmiş AA'nın adımları (Şekil 2) detaylı olarak verilmiştir.

1 Rastgele çözümlerle başlangıç arı popülasyonunu oluştur
2 Popülasyonunun uygunluğunu değerlendir
3 Tekrarla
4 Elit arıları seç
5 Komşuluk araması için bölgeleri seç
6 Arıları seçilen bölgelere gönder ve uygunluklarını değerlendir
7 Her bir bölgedeki en iyi uygunluk değerine sahip arıyı seç
8 Kalan arıları rastgele arama için ata ve uygunluklarını değerlendir
9 Durdurma kriteri sağlanana kadar

Şekil 1. Arı Algoritması'nın Temel Adımları

Pham vd. (2006a)'da bahsedildiği gibi temel AA birçok parametre içermektedir: kâşif arı sayısı (n), ziyaret edilen n nokta içinden seçilen bölge sayısı (m), seçilen m bölge içindeki en iyi bölge sayısı (e), en iyi e bölgeye gönderilen arı sayısı (nep), kalan (m-e) bölgeye gönderilen arı sayısı (nsp), bölge boyutu (ngh) ve durdurma kriteri. Algoritma n adet kâşif arının araştırma uzayına rastgele yerleştirilmesi ile başlar. Kâşif arılarca ziyaret edilen noktaların uygunlukları 2. adımda değerlendirilir. 4. adımda en iyi uygunluk değerine sahip arılar elit arılar olarak, bu arılara ait bölgeler de komşuluk araması için seçilir. 5 ve 6. adımlarda seçilen arıların komşuluğunda araştırma başlar ve daha umut verici çözümleri temsil eden en iyi e bölgeye, seçilen diğer bölgelere göre daha fazla arı gönderilerek daha detaylı arama yapılır. 7 nolu adımda yeni popülasyonun oluşturulması için her bölgedeki en iyi uygunluk değerine sahip arı seçilir. 8 nolu adımda popülasyondaki diğer arılar (n-m) yeni potansiyel çözümler elde etmek için rastgele olarak araştırma uzayına atanırlar. Her bir iterasyonun sonunda yeni popülasyon iki parçadan oluşacaktır: seçilen her bir bölgenin temsilcileri ve rastgele arama yapan kâşif arılar (Pham vd., 2006a,b). Algoritma durdurma kriteri sağlanana kadar devam ettirilir.

2.1 Genelleştirilmiş Atama Problemi

GAP, en basit tanımıyla kapasite kısıtları altında işlerin ajanlara atanması problemidir. Problemin amacı atamalar sonucu oluşan toplam maliyeti minimize etmek olup her iş sadece bir ajan tarafından gerçekleştirilebilmektedir. GAP'nin yerleştirme problemleri, araç rotalama, grup teknolojisi, çizelgeleme gibi birçok uygulama alanı bulunmaktadır. Fisher vd. (1986) tarafından ispatlandığı gibi GAP NP-zor bir yapıya sahiptir ve bu nedenle yazında optimuma yakın çözümler bulmaya yönelik birçok çalışma yapılmıştır. Osman (1995) GAP'nin çözümü için benzetimli tavlama ve tabu arama algoritmalarını geliştirmiştir. Chu ve Beasley (1997) uygunluk ve optimaliteyi aynı anda geliştirmeyi amaçlayan bir genetik algoritma sunmuşlardır. Farklı değişken derinlik arama algoritmaları (Racer ve Amini, 1994; Yagiura vd., 1998, 1999), çıkarım zinciri tabanlı tabu arama algoritmaları

(Laguna vd., 1995; Diaz ve Fernandez, 2001; Yagiura vd., 2004), yol birleştirme yaklaşımları (Alfandari vd., 2001, 2002, 2004; Yagiura vd., 2001, 2002, 2006), açgözlü rastgele adaptif sezgiseline dayalı maks-min karınca sistemi (Lourenço ve Serra, 2002), karınca koloni optimizasyonu (Randall, 2004) yakın zamanda GAP için çözümler bulmaya yönelik olarak geliştirilmiş diğer sezgisel yöntemlerdir. GAP'ne ait tamsayılı doğrusal programlama modeli aşağıdaki gibidir:

$$\sum_{i=1}^{n} a_{ij} x_{ij} \le b_j \qquad \forall j, \quad 1 \le j \le m$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad \forall i, \quad 1 \le i \le n$$

$$x_{ij} \in \{0,1\} \qquad 1 \le i \le n \quad \forall i, \quad 1 \le j \le m \quad \forall j$$

$$k.a.,$$

$$enk \qquad \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij}$$

Modelde yer alan parametrelerden c_{ij} i işini j ajanına atama maliyetini, a_{ij} i işinin j ajanına atanması sonucu oluşan kaynak kullanım miktarını, b_j j ajanının kapasitesini göstermekte olup, I işler kümesini ($i=1,\ldots, n$), J ajanlar kümesini ($j=1,\ldots,m$) temsil etmektedir. x_{ij} ise modelin tek değişkeni olup i işi j ajanına atanmışsa 1, aksi takdirde 0'dır. Amaç fonksiyonu toplam atama maliyetini minimize ederken ilk kısıt kümesi ajanların kapasiteleriyle ilişkili olup ikinci kısıt kümesi her bir işin tek bir ajana atanmasını garantilemektedir.

2.2 Arı Algoritması

GAP'ne uygun çözümler bulabilmeyi amaçlayan Geliştirilmiş AA'nın en önemli adımları Şekil 2'de verilmiştir. Algoritmada kullanılan terimler aşağıdaki gibidir:

s	Kâşif arılar ($s=1\ldots S$)
p	Görevli arılar ($p=1\ldots P$)
e	En iyi görevli arı sayısı
nep	Her bir *e* adet görevli arı için kullanılan izci arı sayısı
nsp	Her bir *P-e* adet görevli arı için kullanılan izci arı sayısı ($nsp<nep$)
maksIter	Maksimum döngü sayısı, durdurma kriteri
makslimit	Gelişme olmaksızın maksimum döngü sayısı
limitsayacı(σ^p)	σ^p için gelişme olmaksızın döngü sayısı
σ^p	*p.* görevli arının çözümü
fit(σ^p)	*p.* görevli arının uygunluk fonksiyonu değeri
σ^{ls}	Yerel arama ile bulunan komşu çözüm (*ls*=komşu çözüm, *en iyi izci*)
σ^{eniyi}	En iyi çözüm
α_j	*j.* ajanın kapasitesinden 1 birim fazla kullanmanın maliyeti

Geliştirilmiş AA, parametrelerin başlangıç değerlerine atanması ile başlar ve genellikle uygun çözümler üreten GRASP algoritması kullanılarak S adet kâşif arı ile başlangıç çözümü oluşturulur. Oluşturulan çözümler kümesindeki *P* adet iyi çözüm görevli arı olarak belirlenir. *P* kümesinden *e* adet çözüm, en iyi çözüm olarak seçilir. Daha detaylı bir komşuluk araması için bu en iyi çözümlere *nep* adet izci arı gönderilir. Daha az sayıda izci arı ise kalan *P-e* adet çözüme gönderilir. Her bir izci arıya ilgili komşuluk yapısı uygulanır. En iyi izci arı orijinal görevli arı ile karşılaştırılır, eğer daha iyiyse görevli arı çözümü güncellenir. Görevli arı çözümleri en iyi çözümle karşılaştırılır, eğer gerekli şartlar sağlanmışsa (*uygun ve bir önceki en iyi çözümden daha iyiyse*) en iyi çözüm güncellenir. Eğer görevli arı çözümü *makslimit* adet iterasyona kadar geliştirilememişse GRASP algoritması kullanılarak yeni bir kâşif arı çözümü oluşturulur. Bütün bu işlemler sonucunda uygun bir çözüm bulunamamışsa α_j değerleri artırılır; en az bir uygun çözüm bulunmuşsa α_j değerleri azaltılır. α_j parametresinin adaptif kontrolünün amacı olursuz çözümleri cezalandırabilmek ve araştırmayı uygun alanlara çekebilmektir. α_j değerlerinin güncellenmesi aşamasının detayları yer kısıtından dolayı verilememiştir ancak; Yagiura vd. (2004) çalışmasından detaylara ulaşılabilir.

GAP'ne özel olarak optimum çözümlerin olursuz çözümlere çok yakın olduğu tespit edildiği için olursuz çözümlere izin verilmesine karar verilmiştir. Diğer

1. Parametreleri başlangıç durumuna getir.
2. GRASP algoritması ile kâşif arı çözümlerini oluştur.
3. Kâşif arı çözümlerinin uygunluk fonksiyonlarını değerlendir (minimizasyon için).

$$fit(\sigma^s) = \sum_{j=1}^{m}\sum_{i=1}^{n} c_{ij}x_{ij} + \left(\sum_{j=1}^{m} \alpha_j \max\left\{0, \sum_{i=1}^{n} a_{ij}x_{ij} - b_j\right\}\right)$$

4. I=0
5. **Tekrarla**
Artan şekilde sırala$_{s=1,\ldots,S}$fit(σ^s) ve en iyi p çözümü görevli arı olarak belirle.
En iyi e adet görevli arıyı seç.
En iyi e adet görevli arının her birine nep adet izci arı ata.
Kalan p-e adet görevli arının her birine nsp adet izci arı ata.
t=0
Tekrarla
k=0
Tekrarla
e adet görevli arıya atanmış her izci arı için komşuluk yapısını uygula.
Eğer fit($\sigma^{komşuçözüm}$) < fit(σ^p) ise $\sigma^p = \sigma^{komşuçözüm}$ **ve** limitsayacı(σ^p)=0,
aksi takdirde limitsayacı(σ^p)=limitsayacı(σ^p)+1
k=*k*+1
Sağlandığı sürece (*k*<*nep*)
k=0
Tekrarla
p-e adet görevli arıya atanmış her izci arı için komşuluk yapısını uygula.
Eğer fit($\sigma^{komşuçözüm}$) < fit(σ^p) ise $\sigma^p = \sigma^{komşuçözüm}$ **ve** limitsayacı(σ^p)=0,
aksi takdirde limitsayacı(σ^p)=limitsayacı(σ^p)+1
k=*k*+1
Sağlandığı sürece (*k*<*nsp*)
Eğer (limitsayacı(σ^p)>makslimit) **ise** GRASP algoritması ile yeni bir kâşif arı çözümü oluştur.
α_j değerlerini güncelle, fit(σ^p)değerlendir.
t=*t*+1
Sağlandığı sürece (*t*<*p*)
En iyi çözümü güncelle, eğer σ^p uygun ve min$_{p=1,\ldots,P}$ fit(σ^p)≤fit(σ^{eniyi}) ise $\sigma^{eniyi}=\sigma^p$
GRASP algoritması ile *s-p* adet yeni kâşif arı çözümleri oluştur.
Sağlandığı sürece (I<maksIter)

Şekil 2. GAP İçin Geliştirilmiş AA Adımları (Özbakır vd., 2009)

taraftan GRASP algoritması ile başlangıç çözümleri oluşturulurken ve komşu çözümler yaratılırken olursuz çözümler üretilebilmektedir. Uygunluk fonksiyonu hesaplanırken kısıtlandırılmış problemi kısıtlandırılmamış hâle dönüştürmek için GAP'ne ait amaç fonksiyonuna bir ceza terimi eklenmiştir. Dolayısıyla GAP üzerine yapılan çalışmaların çoğunda olduğu gibi olursuz alanlarda da araştırmaya izin verilmiş; ancak, olursuz çözümler, olursuzluk derecesine göre cezalandırılmıştır.

2.2.1 Arı Kolonisinin Oluşturulması

Başlangıç arı kolonisi GRASP algoritması (Lourenço ve Serra, 2002) kullanılarak oluşturulmuştur. GRASP algoritması ile her bir adımda atanacak bir sonraki iş seçilerek, seçilen işin atanacağı ajan belirlenir. Bütün işler bir ajana atanana kadar bu iki adım tekrarlanır. GRASP algoritmasında gerçekleştirilen seçim işlemi bir olasılık fonksiyonuna bağlı olarak yapılmakta ve bu fonksiyon her iterasyon sonunda iyi çözümlere göre güncellenmektedir. GRASP algoritmasının temel adımları Şekil 3'te özetlenmiştir.

1. $S_j = \emptyset$, $\forall j=1,\ldots,m$ olsun (S_j=j ajanına atanan işler kümesi)
2. Her bir iş için bütün ajanları içeren bir L_i ajanlar listesi oluştur, başlangıçta $L_i=\{1,\ldots,m\}$ $\forall i$ olsun.
3. İşlerin herhangi bir sırasını al, i=1.
4. **Tekrarla**
4.1 L_i listesindeki bütün ajanlar için aşağıdaki olasılık fonksiyonunu hesapla ve i işinin atanacağı ajanı bu olasılık değerine göre rastgele seç (seçilen ajan j^* olsun):

$$p_{ij} = \frac{b_j / a_{ij}}{\sum_{l \in L_i} b_l / a_{il}}, j \in L_i$$

(Minimum maliyete sahip ajanın seçilme olasılığı daha yüksektir).
4.2 i işini j^*:$S_j^*=S_j^* \cup \{i\}$ ajanına ata.
$i=i+1$ olsun ve eğer $\sum_{i \in S_{j^*}} a_{ij^*} > b_{j^*}$ ise j^* ajanını bütün listelerden sil. 4 nolu adımı tekrarla.
5. **Bütün işler atanana kadar**

Şekil 3. GRASP Algoritması Adımları

3. FARKLI KOMŞULUK YAPILARI

Bu çalışmada AA üzerinde farklı komşuluk yapılarının performansı değerlendirilmiştir. GAP'ne uygun olan dört tip komşuluk yapısı belirlenmiştir:

Kaydırma: Kaydırma komşuluk yapısında bir iş atanmış olduğu ajandan çıkarılarak yeni bir ajana atanmaktadır (i işi j ajanından çıkarılarak w ajanına atanır $w \neq j$). Şekil 4'te 2 nolu iş 2. ajandan çıkarılarak 1. ajana atanmıştır.

Değiştirme: Bir diğer komşuluk yapısı olan değiştirme yönteminde ise iki işin atamasının yer değiştirilmesi gerçekleştirilir (orijinal çözümde i işi j ajanına, k işi w ajanına atanmış iken, i işinin ataması w ajanına, k işinin ataması da j ajanına olacak şekilde değiştirilir). Şekil 5'te 2 ve 4 nolu işlerin ajan atamaları yer değiştirilmiştir.

Çift kaydırma: İki kaydırma hareketinin gerçekleştirildiği çift kaydırma komşuluk yapısı, çıkarım zinciri komşuluğunun özel bir biçimidir. Bu komşuluk yapısı Yagiura vd. (2004) çalışmasında geliştirilen çift kaydırma mekanizmasının basitleştirilmiş halidir (i işi j ajanından çıkarılarak w ajanına atanır $w \neq j$, w ajanından k işi çıkarılır ve q ajanına atanır $q \neq w$). Şekil 6'da 2 nolu iş 2. ajandan çıkarılarak 1. ajana atanmış ve 1. Ajandaki 3 nolu iş ise 3. ajana atanmıştır.

Çıkarım Zinciri: Komşu bir çözüm, uzunluğu zincir uzunluğu olarak belirtilen çok sayıda kaydırma hareketlerinin gerçekleştirilmesiyle elde edilir. Zincir

Ajan-1	Ajan-2	Ajan-3	Ajan-4
İş-5	İş-2	İş-4	İş-10
İş-3	İş-6	İş-1	İş-9
	İş-8	İş-7	

Ajan-1	Ajan-2	Ajan-3	Ajan-4
İş-2	İş-6	İş-4	İş-10
İş-5	İş-8	İş-1	İş-9
İş-3		İş-7	

Şekil 4. Kaydırma Komşuluk Yapısı

Ajan-1	Ajan-2	Ajan-3	Ajan-4
İş-5	İş-2	İş-4	İş-10
İş-3	İş-6	İş-1	İş-9
	İş-8	İş-7	

Ajan-1	Ajan-2	Ajan-3	Ajan-4
İş-5	İş-4	İş-2	İş-10
İş-3	İş-6	İş-1	İş-9
	İş-8	İş-7	

Şekil 5. Değiştirme Komşuluk Yapısı

Ajan-1	Ajan-2	Ajan-3	Ajan-4
İş-5	İş-2	İş-4	İş-10
İş-3	İş-6	İş-1	İş-9
	İş-8	İş-7	

Ajan-1	Ajan-2	Ajan-3	Ajan-4
İş-5	İş-6	İş-4	İş-10
İş-2	İş-8	İş-1	İş-9
		İş-7	
		İş-3	

Şekil 6. Çift Kaydırma Komşuluk Yapısı

uzunluğuna bağlı olarak içerisinde değiştirme, kaydırma ve çift kaydırma komşuluklarını barındırmaktadır (zincir uzunluğu=3 için: *i* işi *j* ajanından çıkarılarak *w* ajanına atanır $w \neq j$, *w* ajanından *k* işi çıkarılır ve *q* ajanına atanır $q \neq w$, *q* ajanından *l* işi çıkarılır ve *t* ajanına atanır $t \neq q$). Şekil 7'de 4 nolu iş 3. ajandan çıkarılarak 2. ajana atanmış, 2. ajana atanmış olan 2 nolu iş 4. ajana atanmış ve son olarak 4. ajandaki 10 nolu iş 1. ajana atanmıştır.

Ajan-1	Ajan-2	Ajan-3	Ajan-4
İş-5	İş-2	İş-4	İş-10
İş-3	İş-6	İş-1	İş-9
	İş-8	İş-7	

Ajan-1	Ajan-2	Ajan-3	Ajan-4
İş-5	İş-6	İş-1	İş-9
İş-3	İş-8	İş-7	İş-2
İş-10	İş-4		

Şekil 7. Çıkarım Zinciri Komşuluk Yapısı (EC-Uzunluğu=3)

4. DENEYSEL ÇALIŞMA

Geliştirilmiş AA, C# dilinde kodlanarak 2.20 GHz CPU, 2.00 GB RAM özelliklere sahip Intel Pentium CoreDuo PC kullanılarak çeşitli test problemlerine uygulanmıştır. Test problemleri OR kütüphanesinden temin edilen (*http://people.brunel.ac.uk/~mastjjb/jeb/orlib/gapinfo.html*) gap-a ve gap-b problem tipleridir. Her bir problem tipi için, her ajan/iş kombinasyonundan (m=5, 10, 20 ve n=100, 200) farklı olarak oluşturulan 12 ayrı problem mevcuttur. Algoritma parametreleri yazında verilen genel öneriler ve yazarların deneyimlerine göre belirlenmiş olup Tablo 1'de gösterildiği gibidir.

Her bir test problemi, geliştirilmiş AA ile 10 kez çalıştırılmış ve elde edilen minimum, ortalama, maksimum ve standart sapma değerleri, optimum CPLEX sonuçları ile birlikte Tablo 2'de; bu verilere ait CPU zamanlarının (bütün CPU zamanları saniye cinsindendir) minimum, ortalama, maksimum ve standart sapma değerleri ise Tablo 3'te verilmiştir. Tablo 2'de min() sütunundaki parantez içindeki değerler algoritmanın 10 kez çalıştırılması sonucunda ilgili minimum değerin kaç kez bulunduğunu ifade etmektedir.

Her bir komşuluk yapısı için test problemlerinin 10 kez çalıştırılması sonucunda elde edilen sonuçların normal dağılıma uymaması, nicel, bağımlı ve parametrik bir yapıya sahip olması nedeniyle komşuluk yapılarının karşılaştırılmasında Wilcoxon Sıralı-Toplam testinin kullanılmasına karar verilmiştir. Her problem tipi için komşuluk yapıları çiftler halinde ele alınarak elde edilen eşitsizlik ilişkisine dayalı Wilcoxon Sıralı-Toplam testi sonuçları Tablo 4'te verilmiştir.

Yapılan analizler neticesinde gapa-1, gapa-3, gapa-4, gapa-6 problemleri için çıkarım zinciri komşuluk yapısının değiştirmeye göre daha üstün olduğu belirlenmiştir. Gapa-2 problemi için önceki problemler için yapılan tespitin yanı sıra çift kaydırma komşuluk yapısının değiştirmeye göre daha iyi performans gösterdiği belirlenmiştir. Gapa-5 problem tipi için kaydırma, çift kaydırma ve çıkarım zinciri komşuluklarıyla elde edilen çözümlerin kendi içinde farklılık göstermemesi sebebiyle Wilcoxon Sıralı-Toplam testi

Tablo 1. Parametre Değerleri

Parametreler	**Başlangıç değerleri**
S (kâşif arı sayısı)	500
P (görevli arı sayısı)	50
e (en iyi görevli arı sayısı)	10
nep (her bir e adet görevli arıya gönderilecek izci arı sayısı)	10
nsp (her bir P-e adet görevli arıya gönderilecek izci arı sayısı)	5
EC-Uzunluğu (çıkarım zinciri komşuluk yapısının uzunluğu)	70
maksIter (maksimum döngü sayısı)	1000
makslimit (gelişme olmaksızın geçirilecek maksimum döngü sayısı)	50
α_j (ceza katsayısının başlangıç değeri)	1

Tablo 2. Farklı Komşuluk Yapılarının Karşılaştırılması

	kaydırma				**çift kaydırma**				**CPLEX 6.5**
	min()	ort	maks	ss	min()	ort	maks	ss	
gapa-1	1698 (10)	1698,0	1698	0,0	1698 (10)	1698,0	1698	0,0	**1698**
gapa-2	3235 (10)	3235,0	3235	0,0	3235 (9)	3235,1	3236	0,3	**3235**
gapa-3	1360 (10)	1360,0	1360	0,0	1360 (10)	1360,0	1360	0,0	**1360**
gapa-4	2623 (10)	2623,0	2623	0,0	2623 (10)	2623,0	2623	0,0	**2623**
gapa-5	1158 (10)	1158,0	1158	0,0	1158 (10)	1158,0	1158	0,0	**1158**
gapa-6	2339 (10)	2339,0	2339	0,0	2339 (10)	2339,0	2339	0,0	**2339**
gapb-1	1855 (1)	1866,2	1879	7,8	1843 (2)	1857,7	1881	12,1	**1843**
gapb-2	3590 (1)	3610,8	3673	24,8	3557 (1)	3565,4	3571	4,4	**3552**
gapb-3	1407 (2)	1409,6	1413	2,3	1407 (3)	1409,6	1412	2,0	**1407**
gapb-4	2859 (1)	2886,2	2919	18,3	2842 (1)	2853,5	2865	6,4	**2827**
gapb-5	1167 (2)	1170,1	1174	2,4	1166 (1)	1169,5	1171	1,7	**1166**
gapb-6	2345 (1)	2349,8	2355	3,4	2343 (2)	2345,4	2348	1,7	**2339**
	değiştirme				**çıkarım zinciri**				
	min()	ort	maks	ss	min()	ort	maks	ss	
gapa-1	1703 (1)	1711,6	1721	5,1	1698 (4)	1699,4	1702	1,4	
gapa-2	3252 (1)	3264,3	3285	10,0	3235 (7)	3235,5	3237	0,8	
gapa-3	1394 (1)	1406,0	1421	9,2	1360 (8)	1360,2	1361	0,4	
gapa-4	2710 (1)	2728,3	2753	16,0	2635 (1)	2665,7	2698	23,4	
gapa-5	1247 (1)	1264,9	1290	12,8	1158 (10)	1158,0	1158	0,0	
gapa-6	2511 (1)	2559,3	2613	32,9	2400 (1)	2442,0	2485	26,7	
gapb-1	1874 (1)	1889,8	1906	10,0	1857 (1)	1888,5	1915	17,4	
gapb-2	3588 (1)	3604,0	3616	9,7	3628 (1)	3648,5	3677	14,4	
gapb-3	1438 (1)	1470,0	1504	20,5	1407 (6)	1407,8	1411	1,3	
gapb-4	2988 (1)	3010,2	3070	24,1	2846 (1)	2865,2	2881	12,2	
gapb-5	1249 (1)	1275,4	1307	19,4	1167 (2)	1168,9	1172	1,5	
gapb-6	2586 (1)	2653,1	2698	37,8	2346 (1)	2352,5	2357	3,5	

yapılamamaştır. Wilcoxon Sıralı-Toplam testinin yapılamadığı ikili karşılaştırmalar için grup ortalamalarına bakıldığında ise gapa-1, gapa-3, gapa-4 ve gapa-6 problemleri için en iyi performans gösteren komşuluk yapısının kaydırma ve çift kaydırma, ikinci en iyi performansa sahip yapının ise çıkarım zinciri olduğu görülmektedir. Gapa-2 problemi için kaydırma komşuluk yapısının en iyi, çift kaydırma komşuluğunun ikinci en iyi ve son olarak çıkarım zinciri komşuluk yapısının üçüncü en iyi performansa sahip olduğu görülmüştür.

Tablo 3. Farklı Komşuluk Yapılarının CPU Bakımından Karşılaştırılması

	kaydırma				**çift kaydırma**			
	min	ort	maks	ss	min	ort	maks	ss
gapa-1	3,23	4,08	4,59	0,48	6,65	96,04	231,81	71,19
gapa-2	8,21	12,69	16,67	2,67	44,81	286,03	923,00	279,89
gapa-3	4,07	5,61	8,95	1,48	5,31	25,56	90,10	32,42
gapa-4	11,71	54,30	141,85	41,46	37,84	168,02	341,89	114,71
gapa-5	5,53	8,13	10,34	1,56	5,42	7,58	10,25	1,26
gapa-6	14,76	31,57	52,23	11,27	54,84	87,68	156,37	31,56
gapb-1	19,21	60,07	103,95	26,48	23,75	111,69	202,09	63,62
gapb-2	6,46	81,50	193,12	61,33	113,59	592,79	1036,93	324,82
gapb-3	12,40	387,89	1106,04	333,10	11,90	123,87	333,17	95,78
gapb-4	17,68	351,59	805,62	289,03	92,25	399,81	904,64	242,25
gapb-5	71,87	421,04	799,14	236,44	17,89	107,01	209,89	82,32
gapb-6	204,12	676,22	1577,96	455,78	103,92	694,48	1162,64	322,97
	değiştirme				**çıkarım zinciri**			
	min	ort	maks	ss	min	ort	maks	ss
gapa-1	32,37	148,41	534,67	146,31	60,59	273,52	436,43	129,78
gapa-2	53,98	166,86	328,20	88,12	180,54	628,99	1119,48	307,40
gapa-3	34,37	164,77	357,71	88,45	73,12	237,13	498,65	131,99
gapa-4	83,40	139,95	214,64	45,53	268,53	653,85	1159,01	322,72
gapa-5	60,00	102,31	170,65	33,69	14,78	68,00	160,04	42,60
gapa-6	101,53	211,02	406,03	93,70	260,25	642,56	1174,50	303,52
gapb-1	40,32	101,61	191,20	45,09	54,85	668,43	1576,18	396,06
gapb-2	66,00	235,14	448,14	108,23	446,62	838,03	1591,14	359,32
gapb-3	49,65	193,90	468,75	123,50	98,53	352,28	683,46	210,83
gapb-4	89,70	155,79	237,20	49,10	934,93	1847,56	2925,28	731,27
gapb-5	53,75	144,96	248,07	58,23	183,31	564,74	929,70	232,63
gapb-6	67,06	161,28	216,82	45,72	452,45	1299,56	2919,26	841,56

Hiçbir ikili karşılaştırmada Wilcoxon Sıralı-Toplam testinin gerçekleştirilemediği gapa-5 problemi için ise kaydırma, çift kaydırma ve çıkarım zinciri komşuluk yapılarının eşit performansa sahip olup değiştirmeden daha üstün olduğu gözlemlenmiştir.

Gapb problemlerini tek tek ele alacak olursak; gapb-1 problem tipi için kaydırma komşuluk yapısının değiştirme ve çıkarım zinciri komşuluk yapılarına göre daha üstün; çift kaydırma komşuluk yapısının ise yine değiştirme ve çıkarım zinciri komşuluk yapılarına göre daha üstün olduğu belirlenmiş; ancak kaydırma ve çift kaydırma komşuluk yapıları arasında ve değiştirme ve çıkarım zinciri komşuluk yapıları arasında anlamlı bir farklılık tespit edilememiştir. Gapb-2 problemleri için çift kaydırma komşuluk yapısının diğer komşuluk yapılarına göre daha iyi performans gösterdiği belirlenmiştir. Gapb-3 ve gapb-5 problemleri için bütün komşuluk yapılarının değiştirme komşuluk yapısından daha üstün olduğu tespit edilmiş ancak; bu komşuluklar arasında anlamlı bir farklılık belirlenememiştir. Gapb-4 problemleri için çift kaydırma ve çıkarım zinciri komşuluk yapılarının kaydırma ve değiştirme komşuluk yapılarından daha iyi olduğu belirlenmiş; ancak çift kaydırma ve çıkarım zinciri komşuluk yapıları arasında anlamlı bir fark tespit edilememiştir. Gapb-6 problemi için bütün komşuluk yapılarının değiştirme komşuluk yapısından daha üstün olduğu tespit edilmiş, diğer taraftan çift kaydırma komşuluk yapısının kaydırma ve çıkarım zincirine göre daha iyi performans gösterdiği belirlenmiştir.

Sonuç olarak ele alınan genelleştirilmiş atama problemleri için "çift kaydırma" komşuluk yapısının en iyi performansa, "kaydırma" ve "çıkarım zinciri" komşuluk yapılarının ikinci en iyi performansa, "değiştirme" komşuluk yapısının ise en düşük performansa sahip olduğu tespit edilmiştir.

Tablo 4. Wilcoxon Rank-Sum Testi ile Komşuluk Yapılarının Değerlendirilmesi

		Kaydırma, Çift kaydırma	Kaydırma, Değiştirme	Kaydırma, Çıkarım zinciri	Çift kaydırma, Değiştirme	Çift kaydırma, Çıkarım zinciri	Değiştirme, Çıkarım zinciri
gapa-1	ortanca-1	-	-	-	-	-	1711,5
	ortanca-2	-	-	-	-	-	1699,5
	p	-	-	-	-	-	0,0002
gapa-2	ortanca-1	-	-	-	3235	3235	3263
	ortanca-2	-	-	-	3263	3235	3235
	p	-	-	-	0,0002	0,4274	0,0002
gapa-3	ortanca-1	-	-	-	-	-	1406
	ortanca-2	-	-	-	-	-	1360
	p	-	-	-	-	-	0,0002
gapa-4	ortanca-1	-	-	-	-	-	2722,5
	ortanca-2	-	-	-	-	-	2664,5
	p	-	-	-	-	-	0,0002
gapa-5	ortanca-1	-	-	-	-	-	-
	ortanca-2	-	-	-	-	-	-
	p	-	-	-	-	-	-
gapa-6	ortanca-1	-	-	-	-	-	2554,5
	ortanca-2	-	-	-	-	-	2445
	p	-	-	-	-	-	0,0002
gapb-1	ortanca-1	1864,5	1864,5	1864,5	1859	1859	1890,5
	ortanca-2	1859	1890,5	1890	1890,5	1890	1890
	p	0,0963	0,0006	0,0052	0,0003	0,0017	0,9097
gapb-2	ortanca-1	3603,5	3603,5	3603,5	3565,5	3565,5	3605
	ortanca-2	3565,5	3605	3646	3605	3646	3646
	p	0,0002	0,7913	0,0022	0,0002	0,0002	0,0002
gapb-3	ortanca-1	1409	1409	1409	1410	1410	1471,5
	ortanca-2	1410	1471,5	1407	1471,5	1407	1407
	p	0,9698	0,0002	0,0539	0,0002	0,0696	0,0002
gapb-4	ortanca-1	2881	2881	2881	2854,5	2854,5	3005,5
	ortanca-2	2854,5	3005,5	2867,5	3005,5	2867,5	2867,5
	p	0,0003	0,0002	0,0073	0,0002	0,0588	0,0002
gapb-5	ortanca-1	1170	1170	1170	1170	1170	1275
	ortanca-2	1170	1275	1169	1275	1169	1169
	p	0,6232	0,0002	0,3258	0,0002	0,3075	0,0002
gapb-6	ortanca-1	2350	2350	2350	2345,5	2345,5	2658,5
	ortanca-2	2345,5	2658,5	2353	2658,5	2353	2353
	p	0,0082	0,0002	0,1212	0,0002	0,0006	0,0002

- ile gösterilen değerler komşuluk yapıları ile elde edilen değerlerin kendi içinde farklılık göstermemesi nedeniyle Wilcoxon Sıralı-Toplam testinin gerçekleştirilemediğini göstermektedir.
α=0,05

5. SONUÇ

Bu çalışmada NP-zor bir problem olan GAP için geliştirilmiş olan Arı Algoritması dört farklı komşuluk yapısı kullanılarak test edilmiş ve bu komşuluk yapılarının performansı incelenmiştir. Yapılan istatistiksel testler sonucunda "çift kaydırma" komşuluk yapısının en iyi, "değiştirme" komşuluk yapısının ise en kötü performansa sahip olduğu belirlenmiştir. Sonraki çalışmalarda komşuluk yapıları ikili ve üçlü olarak değerlendirilecek ve bu kombinasyonların performansa etkisi analiz edilecektir.

TEŞEKKÜR

Adil Baykasoğlu Türkiye Bilimler Akademisine (TÜBA) akademik çalışmalarına sağladığı destekten dolayı teşekkür eder.

KAYNAKÇA

1. Alfandari, L., Plateau, A., Tolla, P. 2001. "A Two-Phase Path Relinking Algorithm for the Generalized Assignment Problem", 4th International Conference of Metaheuristics, Porto, Portugal, 175-179.
2. Alfandari, L., Plateau, A., Tolla, P. 2002. "A Two-Phase Path Relinking Algorithm for the Generalized Assignment Problem", Teknik Rapor No: 378, CEDRIC, CNAM.
3. Alfandari, L., Plateau, A., Tolla, P. 2004. "A Path Relinking Algorithm for the Generalized Assignment Problem", Metaheuristics: Computer Decision-Making, Hazırlayanlar: Resende, M.G.C., Sousa, J.P., Kluwer Academic Publishers, Boston, 1-17.
4. Baştürk, B., Karaboğa, D. 2006. "An Artificial Bee Colony (ABC) Algorithm for Numeric Function Optimization", IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA.
5. Baykasoğlu, A. 2001. "Goal Programming Using the Multiple Objective Tabu Search", Journal of Operational Research Society, 52(12), 1359-1369.
6. Baykasoğlu, A. 2006. "Applying the Multiple Objective Tabu Search to Continuous Optimization Problems with a Simple Neighborhood Strategy", International Journal for Numerical Methods in Engineering, 65, 406-424.
7. Baykasoğlu, A., Özbakır, L., Tapkan, P. 2007. "Artificial Bee Colony Algorithm and its Application to Generalized Assignment Problem", Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Hazırlayanlar: Chan, F.T.S., Tiwari, M.K., I-Tech Education and Publishing, Vienna, Austria, 113-144.
8. Chong, C.S., Low, M.Y.H., Sivakumar, A.I., Gay, K.L. 2006. "A Bee Colony Optimization Algorithm to Job Shop Scheduling", 37th Winter Simulation, Monterey, California, 1954-1961.
9. Chu, P.C., Beasley, J.E. 1997. "A Genetic Algorithm for the Generalized Assignment Problem", Computers Operations Research, 24, 17-23.
10. Diaz, J.A., Fernandez, E. 2001. "A Tabu Search Heuristic for the Generalized Assignment Problem", European Journal Operational Research, 132, 22-38.
11. Fisher, M.L., Jaikumar, R., Van Wassenhove, L.N. 1986. "A Multiplier Adjustment Method for the Generalized Assignment Problem", Management Science, 32, 1095-1103.
12. Karaboğa, D., Baştürk, B. 2007. "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", Journal of Global Optimization, 39(3), 459-471.
13. Laguna, M., Kelly, J.P., Gonzalez-Velarde, J.L., Glover, F. 1995. "Tabu Search for the Multilevel Generalized Assignment Problem", European Journal of Operational Research, 82, 176-189.
14. Lourenço, H.R., Serra, D. 2002. "Adaptive Search Heuristics for the Generalized Assignment Problem", Mathware and Soft Computing, 9. 209-234.
15. Lucic, P. 2002. "Modeling Transportation Problems Using Concepts of Swarm Intelligence and Soft Computing", Doktora Tezi, Civil Engineering, Faculty of the Virginia Polytechnic Institute and State University.
16. Lucic, P., Teodorovic, D. 2001. "Bee system: Modeling Combinatorial Optimization Transportation Engineering Problems by Swarm Intelligence", 4th Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, 441-445.
17. Lucic, P., Teodorovic, D. 2002. "Transportation Modeling: An Artificial Life Approach", International Conference on Tools with Artificial Intelligence, 216-223.
18. Lucic, P., Teodorovic, D. 2003a. "Computing with Bees: Attacking Complex Transportation Engineering Problems", International Journal on Artificial Intelligence Tools, 12(3), 375-394.
19. Lucic, P., Teodorovic, D. 2003b. "Vehicle Routing Problem with Uncertain Demand at Nodes: The Bee System and Fuzzy Logic Approach", Fuzzy Sets in Optimization, Hazırlayan: Verdegay, J.L., Springer-Verlag, Berlin Heidelberg, 67-82.
20. Osman, I.H. 1995. "Heuristics for the Generalized Assignment Problem: Simulated Annealing and Tabu Search Approaches", OR Spektrum, 17, 211-225.
21. Özbakir, L., Baykasoğlu, A., Tapkan, P. 2009. "Bees Algorithm for Generalized Assignment Problem", Applied Mathematics and Computation, doi:10.1016/j.amc.2009.11.018
22. Pham, D.T., Koç, E., Ghanbarzadeh, A., Otri, S., Rahim, S., Zaidi, M. 2006a. "The Bees Algorithm - A Novel Tool for Complex Optimisation Problems", 2nd International

Virtual Conference on Intelligent Production Machines and Systems, 454-461.

23. Pham, D.T., Otri, S., Ghanbarzadeh, A., Koç, E. 2006b. "Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition", International Conference on Information and Communication Technologies, Damascus, Syria, 1624-1629.
24. Pham, D.T., Koç, E., Ghanbarzadeh, A., Otri, S. 2006c. "Optimisation of the Weights of Multi-Layered Perceptrons Using the Bees Algorithm", 5th International Symposium on Intelligent Manufacturing Systems, Sakarya, Turkey, 38-46.
25. Pham, D.T., Soroka, A.J., Ghanbarzadeh, A., Koç, E., Otri, S., Packianather, M. 2006d. "Optimising Neural Networks for Identification of Wood Defects Using the Bees Algorithm", IEEE International Conference on Industrial Informatics, Singapore, 1346-1351.
26. Pham, D.T., Ghanbarzadeh, A., Koç, E., Otri, S. 2006e. "Application of the Bees Algorithm to the Training of Radial Basis Function Networks for Control Chart Pattern Recognition", 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, Ischia, Italy, 711-716.
27. Pham, D.T., Muhamad, Z., Mahmuddin, M., Ghanbarzadeh, A., Koç, E., Otri, S. 2007a. "Using the Bees Algorithm to Optimise a Support Vector Machine for Wood Defect Classification", Innovative Production Machines and Systems Virtual Conference, Cardiff, UK.
28. Pham, D.T., Afify, A., Koç, E. 2007b. "Manufacturing Cell Formation Using the Bees Algorithm", Innovative Production Machines and Systems Virtual Conference, Cardiff, UK.
29. Pham, D.T., Koç, E., Lee, J.Y., Phrueksanant, J. 2007c. "Using the Bees Algorithm to Schedule Jobs for a Machine", 8th International Conference on Laser Metrology, CMM and Machine Tool Performance, Euspen, UK, 430-439.
30. Pham, D.T., Castellani, M., Ghanbarzadeh, A. 2007d. "Preliminary Design Using the Bees Algorithm", 8th International Conference on Laser Metrology, CMM and Machine Tool Performance, Euspen, UK, 420-429.
31. Pham, D.T., Otri, S., Afify, A., Mahmuddin, M., Al-Jabbouli, H. 2007e. "Data Clustering Using the Bees Algorithm", 40th CIRP International Manufacturing Systems Seminar, Liverpool, UK.
32. Pham, D.T., Soroka, A.J., Koç, E., Ghanbarzadeh, A., Otri, S. 2007f. "Some Applications of the Bees Algorithm in Engineering Design and Manufacture", International Conference on Manufacturing Automation, Singapore.
33. Pham, D.T., Ghanbarzadeh, A. 2007. "Multi-Objective Optimisation Using the Bees Algorithm", Innovative Production Machines and Systems Virtual Conference, Cardiff, UK.
34. Quijano, N., Passino, K.M. 2007. "Honey Bee Social Foraging Algorithms for Resource Allocation Theory and Application", Conference of American Control, New York City, USA.
35. Racer, M., Amini, M.M. 1994. "A Robust Heuristic for the Generalized Assignment Problem", Annals of Operations Research, 50, 487-503.
36. Randall, M. 2004. "Heuristics for Ant Colony Optimisation Using the Generalised Assignment Problem", IEEE Congress on Evolutionary Computation, Portland, Oregon, 2, 1916-1923.
37. Teodorovic, D., Dell'Orco, M. 2005. "Bee Colony Optimization - A Cooperative Learning Approach to Complex Transportation Problems", Advanced OR and AI Methods in Transportation, 51-60.
38. Wedde, H. F., Farooq, M., Zhang, Y. 2004. "BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior", Ant Colony, Optimization and Swarm Intelligence, Lecture Notes in Computer Science, Hazırlayan: Dorigo, M., 3172, Springer Berlin, 83-94.
39. Yagiura, M., Yamaguchi, T., Ibaraki, T. 1998. "A Variable-Depth Search Algorithm with Branching Search for the Generalized Assignment Problem", Optimization Methods and Software, 10, 419-441.
40. Yagiura, M., Yamaguchi, T., Ibaraki, T. 1999. "A Variable-Depth Search Algorithm for the Generalized Assignment Problem", Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Hazırlayanlar: Vob, S., Martello, S., Osman, I.H., Roucairol, C., Kluwer Academic Publishers, Boston, 459-471.
41. Yagiura, M., Ibaraki, T., Glover, F. 2001. "An Effective Metaheuristic Algorithm for the Generalized Assignment Problem", IEEE International Conference on Systems, Man, and Cybernetics, Tucson, Arizona, 242.
42. Yagiura, M., Ibaraki, T., Glover, F. 2002. "A Path Relinking Approach for the Generalized Assignment Problem", International Symposium on Scheduling, 105-108.
43. Yagiura, M., Ibaraki, T., Glover, F. 2004. "An Ejection Chain Approach for the Generalized Assignment Problem", INFORMS Journal on Computing, 16(2), 131-151.
44. Yagiura, M., Ibaraki, T., Glover, F. 2006. "A Path Relinking Approach with Ejection Chains for the Generalized Assignment Problem", European Journal of Operational Research, 169, 548-569.
45. Yang, X.S. 2005. "Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms", IWINAC'2005, LNCS 3562, Hazırlayanlar: Yang, J.M., Alvarez, J.R., Springer-Verlag, Berlin Heidelberg, 317-323.