



KRİL SÜRÜSÜ ALGORİTMASI İLE ATÖLYE ÇİZELGELEME (JOB SHOP SCHEDULING WITH KRILL HERD ALGORITHM)

İlker GÖLCÜK¹, Adil BAYKASOĞLU², F. Selen MADENOĞLU³

ÖZET/ABSTRACT

Krill sürüsü algoritması gerçek hayat problemlerini çözmek amacıyla yakın dönemde literatüre kazandırılmış sürü temelli metasezgisel algoritmalar biridir. Algoritmanın performansı literatürde sürekli-sayı değişkenlere sahip doğrusal olmayan optimizasyon problemleri üzerinde denenmiştir. Bu çalışmada krill sürüsü algoritmasının performansı literatürde ilk kez kombinatoriyal optimizasyon problemlerinden biri olan atölye tipi çizelgeleme problemleri üzerinde test edilmiştir. Atölye tipi çizelgeleme problemleri, diğer zor kombinatoriyal optimizasyon problemlerini temsil eden önemli bir problem türü olduğundan, bu çalışma algoritmanın diğer kombinatoriyal problemlerdeki olası performansı hakkında ön bilgiler vermektedir.

Krill herd algorithm is a recently proposed swarm based metaheuristic algorithm for solving real life problems. In the literature, the performance of the algorithm has been tested on non-linear continuous optimization problems. In this study, the performance of the krill herd algorithm is tested on job shop scheduling problems, which is one of combinatorial optimization problems, for the first time in the literature. Job shop scheduling problems are one of the most complex and important problems with representative characteristics to other hard combinatorial optimization problems. This study provides some idea about krill herd algorithm's possible performance for solving other combinatorial optimization problems.

ANAHTAR KELİMELER/KEYWORDS

Krill sürüsü algoritması, Atölye çizelgeleme problemi, Metasezgisel algoritmalar, Sürü zekası

Krill herd algorithm, Job shop scheduling problem, Metaheuristic algorithms, Swarm intelligence

¹ Dokuz Eylül Üniversitesi, Müh. Fak., Endüstri Müh. Böl., İZMİR; e-posta: ilker.golcuk@deu.edu.tr

² Dokuz Eylül Üniversitesi, Müh. Fak., Endüstri Müh. Böl., İZMİR; e-posta: adil.baykasoglu@deu.edu.tr

³ Dokuz Eylül Üniversitesi, Müh. Fak., Endüstri Müh. Böl., İZMİR; e-posta: selen.madenoglu@deu.edu.tr

1. GİRİŞ

Çizelgeleme problemleri üretim, dağıtım, ulaştırma, inşaat, mühendislik, yönetim gibi alanlarda sıklıkla karşılaşılan ve endüstri mühendislerinin yoğunluklu çalıştığı alanlardan bir tanesidir. Özellikle üretim ve imalat alanlarındaki çizelgeleme problemleri oldukça karmaşık yapıda olup, klasik optimizasyon yaklaşımları bu problemleri çözmeye çok yetersiz kalmaktadır. Bu nedenle, yöneylem araştırması, yapay zeka, üretim mühendisliği ve endüstri mühendisliği gibi araştırma disiplinlerinde çalışan araştırmacıların çizelgeleme problemlerine olan ilgisi hep yüksek seviyelerde olagelmıştır.

Çizelgeleme problemleri, üretim yönetimi alanında karşılaşılan en önemli problemlerden biridir. Çizelgeleme problemlerinde temel amaç, eldeki makinelerde işlem görecektir olan işlerin en uygun sıralamasını bularak işlerin maximum bitirilmesi zamanlarını en küçükmektir. Atölye çizelgeleme problemleri, akış tipi çizelgeleme problemleriyle birlikte en çok çalışılmış çizelgeleme problemlerinin başında gelmektedir. Akış tipi çizelgeleme problemlerinde işlerin uğrayacağı makinelerin sırası her parça için aynıyken, atölye tipi çizelgeleme problemlerinde her iş farklı makine rotalarını izlemektedir. Çizelgeleme problemlerinde iş ve makine sayıları arttıkça, birçok çizelge içerisinden en iyi çizelgenin seçilmesi oldukça karmaşık bir hal almaktadır (Anandaraman, 2011). Bunun başlıca sebebi, çözüm uzayının üstel bir biçimde büyümesi ve algoritmaların hesaplama zamanlarının dramatik biçimde artmasıdır. Örneğin n iş ve m makineden oluşan atölye çizelgeleme probleminde tüm olası çizelgelerin sayısı $(n!)^m$ olmaktadır (Luh ve Chueh, 2009). Küçük boyutlu problemlerde dal-sınır algoritması gibi yöntemlerle optimal çizelgeleri bulmak mümkün olabilmektedir (Brucker vd., 1994). Ancak problem büyüklüğü arttıkça, optimal çizelgelerin polinom zamanda elde edilmesi imkansız hale gelmektedir (Brucker vd., 1994). Bu sebeple atölye çizelgeleme problemlerinin çözümünde birçok sezgisel ve metasezgisel yöntem önerilmiştir. Değişen darboğaz sezgiseli, dal-sınır algoritması, tavlama benzetimi algoritması, memetik evrim algoritması, tabu arama paralel, GRASP gibi çok sayıda yöntem geliştirilmiştir (Adams vd., 1988; Brucker vd., 1994; Baykasoğlu, 2002; Hasan vd., 2009; Nowicki ve Smutnicki, 1996; Aiex vd. 2003). Yakın zaman önce yanlı rastgele anahtar genetik algoritması yöntemiyle birçok test problemindeki en iyi çözümü güncellemişlerdir (Gonçalves ve Resende, 2014). Atölye çizelgeleme problemine ait literatür derlemeleri ve makalelerinden bulunabilir (Jain ve Meeran, 1999; Potts ve Strusevich, 2009).

Atölye çizelgeleme probleminde olduğu gibi, birçok gerçek hayat probleminde metasezgisel algoritmalar oldukça başarılı sonuçlar vermektedir. Araştırmacıların yoğun ilgisini çeken bu alanda, doğada gözlemlenen sürü hareketleri ve evrimsel mekanizmaları temel alan bir çok yeni algoritma önerilmektedir. Kril sürüsü algoritması, bu motivasyonlar ışığında, antartikada yaşayan kril canlılarının beslenme davranışlarından esinlenilerek kısa bir süre önce geliştirilmiş sürü temelli bir metasezgisel algoritmadır (Gandomi ve Alavi, 2012). İlgili literatür incelendiğinde, kril sürüsü algoritmasının bir çok farklı problemi çözmeye kullanıldığı gözlemlenmektedir. Gandomi ve Alavi lineer olmayan test fonksiyonları üzerinde kril sürüsü algoritmasının performansını test etmişlerdir (Gandomi ve Alavi, 2012). Singh ve Sood kril sürüsü algoritması temelli bir kümeleme algoritması önermişlerdir (Singh ve Sood; 2013). Wang vd., global optimizasyon problemleri için tavlama benzetimi temelli kril sürüsü algoritması geliştirmişlerdir (Wang vd., 2013). Wang vd. kaotik parçacıklı kril sürüsü algoritması geliştirerek global nümerik optimizasyon problemlerine uygulamışlardır (Wang vd., 2013). Roy ve Paul, kril sürüsü algoritmasıyla optimum güç akışını modellemişlerdir (Roy ve Paul, 2014). Global nümerik optimizasyon problemleri çözümündeki performansı arttırmak amacıyla kril sürüsü algoritmasına mutasyon operatörü eklenmiştir (Wang vd., 2014). Benzer biçimde, kril sürüsü algoritmasıyla biyocoğrafya algoritmasının göç operatörlerini

birleştirilerek hibrit bir metasezgisel yaklaşım önerilmiştir (Wang vd., 2014). Ayrıca kril sürüsü algoritmasının performansını geliştirmek için Levy-Flight ve Stud kril sürüsü algoritmaları geliştirilmiştir (Wang vd., 2013; Wang vd., 2014).

Kril sürüsü algoritmasının bir çok mühendislik problemi çözümünde etkili olabileceği yapılan literatür analiziyle belirlenmiştir. Ancak algoritma henüz herhangi bir kombinatoriyal optimizasyon probleminde uygulanmamıştır. Çizelgeleme ve özellikle atölye tipi çizelgeleme problemleri, tipik kombinatorial optimizasyon problemi olup, kesin yöntemlerle çözüm elde etmek oldukça zordur ve NP-zor sınıfına girmektedir (Baker, 1974). Bu çalışmada kril sürüsü algoritmasının performansı literatürde ilk kez çeşitli atölye tipi çizelgeleme test problemleri üzerinde denenmiş ve iyi sonuçlar elde edilmiştir. İkinci bölümde atölye tipi çizelgeleme problemi üzerinde durulacak, üçüncü bölümde kril sürüsü algoritması açıklanacak, dördüncü bölümde deney sonuçlarına yer verilecektir. Sonuçlar ve gelecek çalışmalara beşinci bölümde değinilmiştir.

2. ATÖLYE ÇİZELGELEME PROBLEMİ

Bir atölye tipi çizelgeleme probleminde her bir işin operasyon sırasını tamamlamak amacıyla m adet makineyi ziyaret ettiği n adet iş ve m adet makineden oluşan bir sistem söz konusudur. Her operasyon o işin tamamlanması için m makineden bir tanesini kullanmak durumundadır. Genel olarak, bir makinede işlem gören bir iş, bir operasyon olarak tanımlanır ve O_{ji} şeklinde ifade edilir (O_{ji} i . makinede işlem gören j . işi temsil etmektedir). Atölye tipi çizelgeleme probleminin temel amacı toplam üretim süresini en küçükleyecek şekilde tüm makineler için uygun çizelgeler üretmektir. İşler kümesi $J = \{1, 2, \dots, n\}$ ile makineler kümesi ise $M = \{1, 2, \dots, m\}$ ile verilmiş olsun. Çizelgelenecek operasyonlar kümesi $O = \{0, 1, \dots, n \times m, n \times m + 1\}$ ile gösterilmektedir. Burada 0 ve $n \times m + 1$ kukla başlangıç ve bitiş operasyonlarını ifade etmektedir. T_j sabit işlem zamanlarını, F_j j . işin bitme zamanı, $A(t)$ t zamanında işlenecek operasyonlar kümesini göstermektedir. e_{jm} değişkeni eğer operasyon j makine m üzerinde işlem göreceyse 1 değerini, aksi durumda 0 değerini almaktadır. Öncelik ilişkilerine dayalı kısıtlar her bir operasyon j nin, önceki tüm öncül işler P_j işlerini bitirdikten sonra çizelgelenmesini garanti eder. Ayrıca operasyon j yalnızca ihtiyaç duyulan makine boşaysa çizelgelenmektedir. Atölye tipi çizelgeleme problemine ait kavramsal model aşağıdaki gibidir:

$$\begin{aligned}
 & \min F_{n \times m + 1} \\
 & \text{s.t.}; F_k \leq F_j - T_j, \quad j = 1, 2, \dots, n \times m + 1; \quad k \in P_j \\
 & \sum_{j \in A(t)} e_{jm} \leq 1, \quad m \in M; \quad t \geq 0 \\
 & F_j \geq 0, \quad j = 1, 2, \dots, n \times m + 1
 \end{aligned} \tag{1}$$

3. KRİL SÜRÜSÜ ALGORİTMASI

Kril sürüsü algoritması, antartik krillerinin beslenme davranışlarından esinlenilerek Gandomi ve Alavi tarafından ortaya konmuş sürü zekasına dayalı bir metasezgisel algoritmadır (Gandomi ve Alavi, 2012).

Krill sürüsü algoritması sürünün başka canlılar tarafından avlandığı ve sürüdeki ortalama kril yoğunluğunun azalarak yemek kaynağından uzaklaşılma durumu başlangıç aşaması olarak

kabul eder. Doğal bir sistemde, her bireyin uygunluk değeri, kril sürüsünün en yüksek yoğunluk noktası ve yemeğin bulunduğu noktalara uzaklığın bir birleşimi şeklinde hesaplanır. Böylece uygunluk, amaç fonksiyonunun değeri olmaktadır. Kril bireylerinin iki boyutlu yüzeyde zamana bağlı pozisyonları üç temel eylem sonunda gerçekleşmektedir (Hofmann vd., 2004):

- Diğer kril bireylerinin sebep olduğu hareket
- Yem arama aktivitesi
- Rassal yayılma

Böylece n boyutlu karar uzayında ifade edilen Lagrange Modeli:

$$\frac{dX_i}{dt} = N_i + F_i + D_i \quad (2)$$

Burada N_i diğer kril bireylerinin sebep olduğu hareketi, F_i yem arama hareketini ve D_i i . kril bireyin fiziksel yayılmasını ifade etmektedir.

3.1. Diğer Kril Bireylerinin Sebep Olduğu Hareket

Her bir kril bireyi için bu hareket aşağıdaki şekilde ifade edilmektedir.

$$N_i^{yeni} = N^{maks} \alpha_i + w_n N_i^{eski} \quad (3)$$

$$\alpha_i = \alpha_i^{lokal} + \alpha_i^{hedef} \quad (4)$$

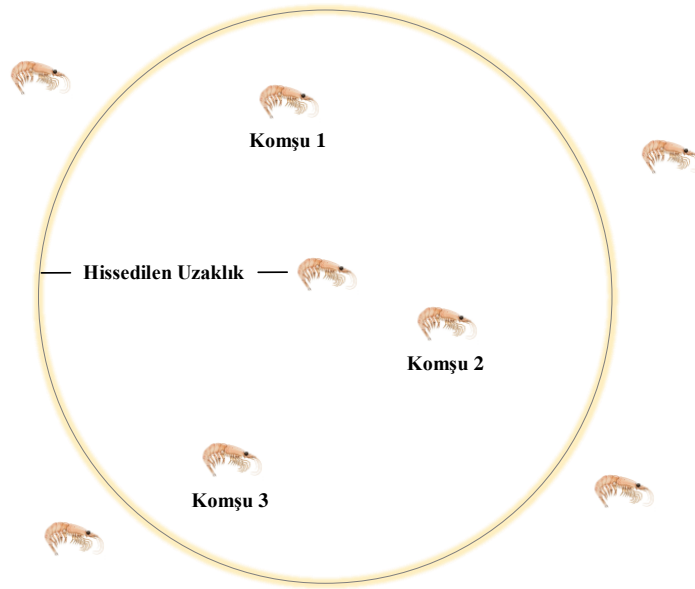
Burada N maksimum sebep olunan hız, 0.01 (m/s) olarak alınmıştır. α_i sebep olunan hareketin doğrultusu, w_n sebep olunan hareketin atalet ağırlığı, N_i^{eski} son sebep olunan hareket, α_i^{lokal} komşu bireyler tarafından sağlanan lokal etkiler ve α_i^{hedef} en iyi kril bireyi tarafından sağlanan hedef yönün etkisidir. Kril bireyin hareketine komşu krillerin etkisi eşitlik 5-7 tarafından formülize edilir:

$$\alpha_i^{lokal} = \sum_{j=1}^{NN} \widehat{K}_{i,j} \widehat{X}_{i,j} \quad (5)$$

$$\widehat{X}_{i,j} = \frac{x_j - x_i}{\|x_j - x_i\| + \varepsilon} \quad (6)$$

$$\widehat{K}_{i,j} = \frac{K_i - K_j}{K^{enkötü} - K^{eniyi}} \quad (7)$$

Burada $K^{enkötü}$ ve K^{eniyi} kril bireylerinin şimdiye kadarki en kötü ve en iyi uygunluk değerlerini ifade etmektedir. K_i i . kril bireyin amaç fonksiyonu değerini temsil ederken, K_j j . komşu bireyin amaç fonksiyonu değeridir. Her kril bireyin pozisyonu X ile ifade edilirken, NN toplam komşu sayısını göstermektedir. Komşu seçim işlemi d_s ile ifade edilen hissedilen uzaklık (sensing distance) baz alınarak Şekil 1'deki gibi yapılmaktadır.



Şekil 1. Hissedilen uzaklık ve komşuluk yapısı (Gandomi ve Alavi, 2012)

$$d_{s,i} = \frac{1}{5N} \sum_{j=1}^N \|X_i - X_j\| \quad (8)$$

Burada $d_{s,i}$ i . kril bireyinin hissedilen uzaklığını gösterirken, N toplam kril bireyi sayısını vermektedir. Sekiz nolu denkleme göre eğer iki kril bireyi arasındaki uzaklık d_s 'den küçükse, kril bireylerinin komşu olduğu sonucu çıkartılmaktadır. En iyi amaç fonksiyonu değerine sahip olan kril bireyinin i . kril bireyi üzerine olan etkisi şu şekilde modellenmektedir:

$$\alpha_i^{hedef} = C^{eniyi} \widehat{K}_{i,eniyi} \widehat{X}_{i,eniyi} \quad (9)$$

C^{eniyi} etki katsayısı olmak üzere

$$C^{eniyi} = 2\left(rassal + \frac{I}{I_{maks}}\right) \quad (10)$$

Burada rastsal $[0,1]$ aralığında rastsal sayı, I döngü sayısı ve I_{maks} maksimum döngü sayısıdır.

3.2. Yem Arama Aktivitesi

Yem arama aktivitesi iki temel kavramla ilişkilidir. Bunlardan ilki yem lokasyonu, ikincisiyse önceki tecrübedir. Bu hareket i . kril bireyi için şu şekilde tanımlanmaktadır:

$$F_i = V_f \beta_i + w_f F_i^{eski} \quad (11)$$

$$\beta_i = \beta_i^{yem} + \beta_i^{eniyi} \quad (12)$$

Burada V_f yem arama hızını göstermektedir ve Price'a dayanarak 0.02 (m/s) olarak alınmıştır (Price, 1989). w_f yem arama hareketinin atalet ağırlığı, F_i^{eski} son yem arama hareketi, β_i^{yem} yem çekiciliği ve β_i^{eni} i . kril bireyinin şuana kadarki en iyi amaç fonksiyonu değerinin etkisidir. Her bir iterasyonda yemek merkezi şu şekilde tanımlanmaktadır:

$$X^{yem} = \frac{\sum_{i=1}^N (1/K_i) X_i}{\sum_{i=1}^N (1/K_i)} \quad (13)$$

Böylece i . kril bireyi için yem çekiciliği şu şekilde ifade edilmektedir:

$$\beta_i^{yem} = C^{yem} \widehat{K}_{i,yem} \widehat{X}_{i,yem} \quad (14)$$

$$C^{yem} = 2 \left(1 - \frac{I}{I_{maks}}\right) \quad (15)$$

i . kril bireyine ait en iyi amaç fonksiyonu değerinin etkisi şu şekilde modellenmektedir:

$$\beta_i^{eni} = \widehat{K}_{i,eni} \widehat{X}_{i,eni} \quad (16)$$

Burada $K_{i,eni}$ daha önceden ziyaret edilmiş en iyi pozisyondur.

3.3. Fiziksel Yayılma

Rassal bir süreç olan fiziksel yayılma şu şekilde formülize edilmektedir.

$$D_i = D^{maks} \left(1 - \frac{I}{I_{maks}}\right) \delta \quad (17)$$

Burada maksimum yayılma hızı D^{maks} olmak üzere, $D^{maks} \in [0.002, 0.010]$ (m/s), ve δ rastsal yönlü vektör ve değerleri $[-1, 1]$ aralığındadır.

3.4. Kril Sürüsü Algoritması Hareket Süreci

Kril bireyinin t ve $t + \Delta t$ zaman aralığında pozisyon vektörü şu şekilde ifade edilmektedir:

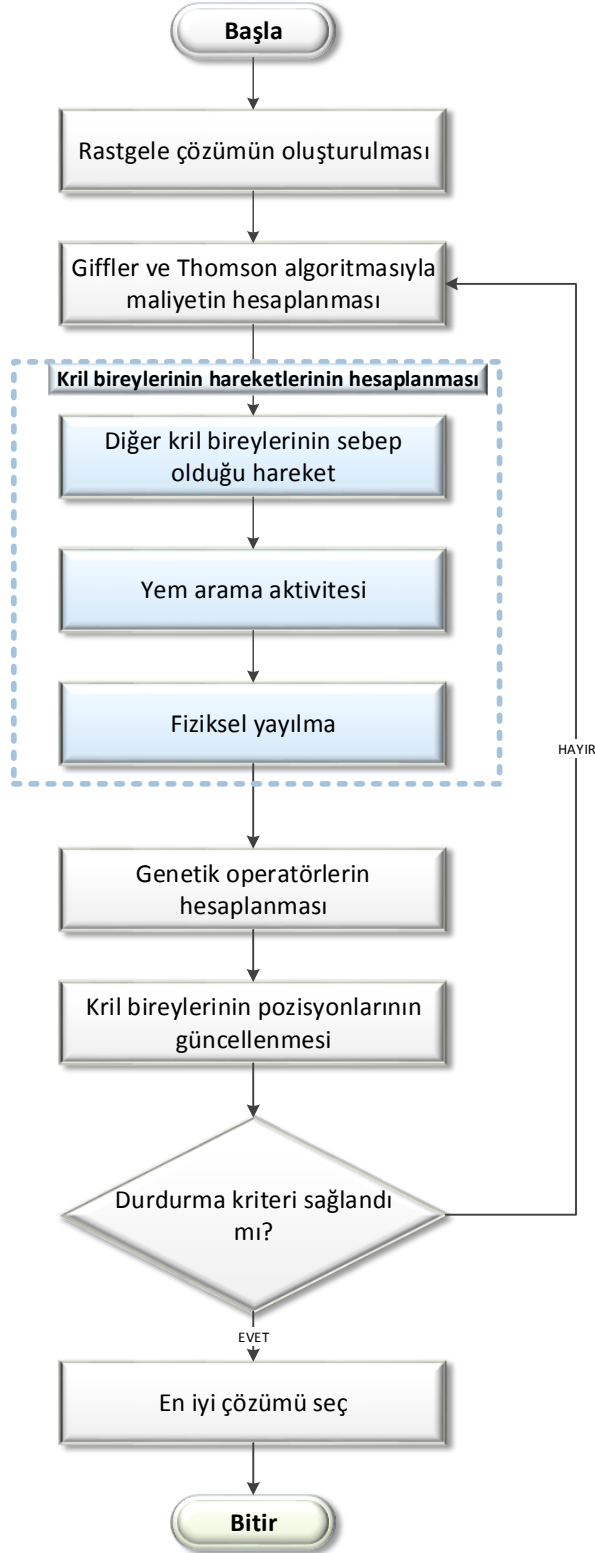
$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt} \quad (18)$$

$$\Delta t = C_t \sum_{j=1}^{NV} UB_j - LB_j \quad (19)$$

Burada NV toplam değişken sayısını, LB_j ve UB_j j . değişkene ait alt ve üst limitleri, C_t sabit bir değeri göstermektedir. C_t değeri 0.5 alınmıştır.

4. KRİL SÜRÜSÜ ALGORİTMASININ UYGULANMASI

Kril sürüsü algoritmasının atölye çizelgeleme problemine uygulanmasını gösteren akış diyagramı Şekil 2’de gösterilmiştir.



Şekil 2. Önerilen çözüm yönteminin akış diyagramı

Önerilen çözüm yönteminde önce rastgele anahtar gösterimi kullanılarak başlangıç çözümleri oluşturulmaktadır. Daha sonra rastgele anahtarlar operasyon sıralarına dönüştürülerek çözümün maliyeti hesaplanmaktadır. Ardından kril sürüsü algoritması adımları sırayla hesaplanmakta ve pozisyon değerleri güncellenmektedir. Bulunan pozisyon değerleri yeniden Giffler ve Thomson algoritması kullanılarak aktif çizelgelere dönüştürülüp maliyeti hesaplanmaktadır. Burada maliyet işlerin tamamlanma süreleridir (Giffler ve Thomson, 1960). Durdurma kriteri sağlanana kadar algoritma çalışmakta ve sonuçlar güncellenmektedir. Önerilen yöntemde en önemli adımlar çözüm gösterimi ve maliyet fonksiyonunun hesaplanmasıdır. Bir sonraki bölümde çözüm gösterimi ve maliyet fonksiyonu hesaplanması ayrıntılı olarak açıklanacaktır.

4.1. Çözüm Gösterimi

Atölye çizelgeleme probleminin metasezgisel algoritmalar kullanılarak çözülmesi konusundaki en önemli adımlardan biri, kullanılacak algoritma ve problemin yapısını dikkate alan en uygun problem gösterimini seçmektir. Literatürde operasyon tabanlı gösterim, iş tabanlı gösterim, tercih listesi tabanlı gösterim, öncelik kuralı tabanlı gösterim, makine tabanlı gösterim ve rastgele anahtar gösterimi gibi bir çok yöntem kullanılmıştır (Cheng vd., 1996). Çözüm gösterimi oluşturulurken, atölye çizelgeleme problemindeki iki sıralama ilişkisi büyük önem arz etmektedir. Bunlar her makinedeki operasyon sıraları ve iş için varolan öncelik ilişkileridir (Gen ve Cheng, 2000). Çözüm gösteriminin bu öncelik ilişkilerini dikkate alarak yalnız uygun çözümler üretmesi büyük önem arz etmektedir. Bu çalışmada temel alınarak rastgele anahtar gösterimi kullanılmıştır (Lin vd., 2010).

Rastgele anahtar gösterimi kril bireylerinin pozisyonlarını sürekli-değişken uzayından kesikli uzaya dönüştürmek amacıyla kullanılmaktadır. Rastgele anahtar uzayındaki vektör reel sayılardan oluşmaktadır. Rastgele anahtar gösteriminde, popülasyonda her bir bireyi temsil eden reel sayı vektörleri, kesikli sayılardan oluşan operasyon permutasyonlarını simüle etmek amacıyla kullanılır. İş sayısının n ve makine sayısının m olduğu bir atölye çizelgeleme probleminde, rastgele anahtar $n \times m$ boyutunda olmaktadır. Popülasyondaki her birey R_j ile ifade edilmekte, R_j ilgili operasyon sırasının ağırlığını göstermektedir. Burada R_j reel sayılardan oluşmakta ve $1 \leq j \leq n \times m$ olmaktadır. Rastgele anahtar gösterimi kullanımında genel olarak dört adım takip edilmiştir. Bunlar rastgele anahtarların oluşturulması, tamsayı serilerinin elde edilmesi, iş indekslerinin oluşturulması ve operasyon sıralarının belirlenmesidir. Bu adımları kısaca açıklayacak olursak;

1. Rastgele anahtar oluşturulması: Rastgele anahtar algoritmanın başlangıcında bir rastgele sayı üreticisi yardımıyla oluşturulmaktadır. Oluşturulan rastgele anahtar, ilerleyen iterasyonlarla beraber algoritmada tanımlanan kril bireylerinin hareketleri boyunca güncellenmektedir.
2. Tamsayı serilerinin elde edilmesi: Rastgele anahtar gösterimi kullanırken öncelikle reel sayılardan oluşan R_j vektörü, küçükten büyüğe doğru sıralanarak tam sayı serileri elde edilir. Tam sayı serileri $(\pi_1, \pi_2, \dots, \pi_k)$ şeklinde gösterilirken, her tamsayı π_k 1 ve $n \times m$ arasında olmaktadır. Her tamsayı serisi π_k dolaylı olarak işin operasyon sırasını göstermektedir.
3. İş indekslerinin oluşturulması: Atölye çizelgeleme probleminin doğası gereği her iş m adet makineden geçmek zorunda olduğu için, bir iş aynı zamanda m adet operasyon için çizelgelenmek zorundadır. Bu özellik dikkate alınarak tam sayı serileri

$(\pi_1, \pi_2, \dots, \pi_k, \dots, \pi_{nm})$, $(\pi_k \bmod n) + 1$ formülü yardımıyla iş indekslerine dönüştürülür. Örneğin 1. işe ait olan operasyonlar $\pi_k \in (n, 2n, \dots, n \times m)$ şeklinde gösterilirken, $\pi_k \in (1, n+1, \dots, (m-1) \times n + 1)$ ifadesi 2. işe ait tüm operasyonları ortaya koymaktadır.

4. Operasyon sıralarının belirlenmesi: İş indeksinde soldan sağa doğru gidildikçe, her bir işin m kez tekrarlandığı gözlemlenmelidir. Böylece, iş indeksi soldan sağa doğru taranırken bir işin i . kez tekrar etmesi, o işin m makine içerisindeki i . operasyonunu göstermektedir. Bu yöntemle her zaman uygun olan çözümler üretilmiş olmaktadır. Operasyon sıraları algoritmada maliyet fonksiyonu hesaplanmasında kullanılmaktadır.

Şekil 3' de, 3 iş ve 2 makineden oluşan bir atölye çizelgeleme problemi için sürekli-değişken biçimindeki rastgele anahtarların, kesikli uzaydaki iş indeksleri ve operasyon sıralarına dönüşümü gösterilmiştir. Başlangıçta elimizde oluşturulan rastgele anahtar $(0.81, 0.91, 0.13, 0.94, 0.63, 0.10)$ şeklinde olsun. Bu sayılar küçükten büyüğe doğru sıralanarak tamsayı serileri elde edilir. Örneğin rastgele anahtar sıralandığında 0.81 değeri 4. sırada yer alırken, 0.91 değeri 5. sırada yer almaktadır.

0.81	0.91	0.13	0.94	0.63	0.10	Rastgele anahtar
4	5	2	6	3	1	Tamsayı serisi
2	3	3	1	1	2	İş indeksleri
O_{21}	O_{31}	O_{32}	O_{11}	O_{12}	O_{22}	Operasyon sıraları

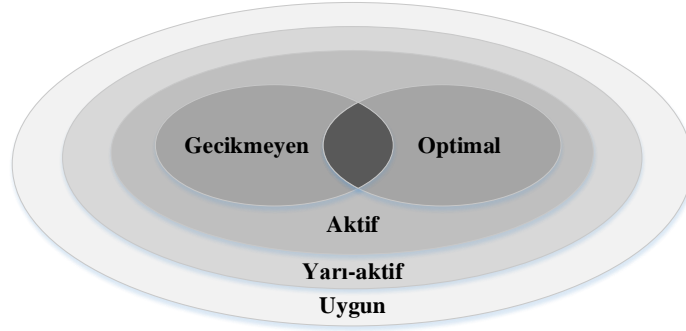
Şekil 3. Rastgele anahtar gösterimi

Elde edilen tamsayı serisi içerisinde 4 ve 1 numaralı tamsayı değerleri 2. işi temsil etmektedir. Çünkü $(4 \bmod 3) + 1 = 2$ ve $(1 \bmod 3) + 1 = 2$ olmaktadır. Benzer şekilde 5 ve 2 numaralı tamsayı serileri 3. işe karşılık gelirken, 6 ve 3 numaralı tamsayı serileri 1. işi göstermektedir. Bulunan iş indeksleri soldan sağa doğru taranarak operasyon sıraları bulunmaktadır. Örneğin 1. iş indeksi olan 2 numara ele alındığında, 2 ilk kez geçtiği için 2. işin 1. operasyonu olup O_{21} şeklinde ifade edilmektedir. Ardından gelen 3 numaralı iş indeksi ilk kez geçtiği için 3. işin 1. operasyonunu temsil eder ve O_{31} şeklinde yazılır. Daha sonra gelen 3 numaralı iş indeksi, daha önce 1 kez 3 numaralı iş indeksi geçtiğinden, 3 numaralı işin 2. operasyonunu temsil eder. Benzer mantıkla tüm operasyon sıraları bulunur ve bulunan operasyon sıraları Giffler ve Thomson algoritmasıyla birlikte maliyet fonksiyonunun hesaplanmasında kullanılır.

4.2. Maliyet Fonksiyonu Hesaplanması

Atölye çizelgeleme problemlerinde dört çeşit çizelgeden bahsedilebilir; bunlar uygun olmayan çizelgeler, yarı aktif çizelgeler, aktif çizelgeler ve gecikmeyen çizelgelerdir (Hong Wei vd., 2008). Uygun olmayan çizelgeler çok fazla atıl zaman içeren çizelgeler olup çizelgedeki operasyonlar ileri kaydırılarak iyileştirilirler. Yarı aktif çizelgeler mümkün olan en erken başlama zamanları dikkate alınarak operasyonların peşi sıra çizelgelenmesiyle oluşturulur (Bierwirth ve Mattfeld, 1999). Yarı aktif çizelgeler fazladan atıl zaman içermezken, diğer işleri geciktirmeyecek şekilde bazı işlerin öne alınmasıyla geliştirilebilirler (Hong Wei

vd., 2008). Aktif çizelgelerde atıl zaman bulunmaz ve herhangi bir iş diğer işleri geciktirmeyecek biçimde daha önce bitirilemez. Gecikmeyen çizelgeler ise aynı zamanda aktif çizelgeler olup, operasyonlar, makinenin atıl kalma süreleri minimize edilecek ve işlenmeye hazır işin olması durumunda makinenin boşta kalması engellenecek bir biçimde çizelgeye yerleştirilir. Bahsedilen çizelgeler arasındaki ilişki Şekil 4' de verilmiştir.



Şekil 4. Çizelgeler arasındaki ilişki (Bierwirth ve Mattfeld, 1999)

Şekil 4'de görüldüğü gibi optimal çizelgeler aynı zamanda aktif çizelgelerdir. Bu sebeple yalnızca aktif çizelgelere odaklanarak, optimal çözümü aktif çizelgeler arasından aramak hesaplamada çok önemli kolaylıklar sağlayacaktır. Aktif çizelgelerin oluşturulması Giffler ve Thomson algoritmasıyla hesaplanabilmektedir.

Bu çalışmada Giffler ve Thomson algoritması aktif çizelgeler oluşturmak amacıyla kril sürüsü algoritması içerisinde kullanılmıştır. Kril algoritması rastsal anahtarlar gösterimi kullanılarak çizelgeleme problemine uygulanmıştır. Rastsal anahtar ile Giffler ve Thomson algoritması içindeki optimal öncelikler kril algoritması ile belirlenmeye çalışılmıştır. Giffler ve Thomson algoritması notasyonu ve adımları şu şekildedir.

Notasyon:

(i, j) : i . makede işlenmesi gereken j . işe ait operasyon

S : Çizelgelenmiş operasyonları içeren kısmı çizelge

Ω : Çizelgelenebilir operasyonlar kümesi

$e_{(i,j)}$: Ω 'ya ait (i, j) operasyonunun en erken başlama zamanı

$p_{(i,j)}$: (i, j) için işlem süreleri

$f_{(i,j)}$: Ω 'ya ait (i, j) operasyonunun en erken bitiş zamanı

$$f_{(i,j)} = e_{(i,j)} + p_{(i,j)}$$

GT Algoritması:

Adım 1: Başlat $S = \emptyset$; Ω öncelik ilişkilerini içeren tüm operasyonları içeren küme.

Adım 2: En erken tamamlanma süresi f^* değerine sahip $(i, j)^*$ operasyonunu Ω içinde bul ve f^* gerçekleştirilebileceği makine m^* kaydet.

Adım 3:

3.1. $e_{(i',j')} < f^*$ olacak şekilde m^* makinesine ihtiyaç duyan $(i', j') \in \Omega$ operasyon kümesini tespit et.

3.2. Adım 3.1. deki operasyon kümesi içerisinde (i, j) operasyonunu en yüksek önceliğe göre seç

3.3. (i, j) 'yi S 'ye ekle.

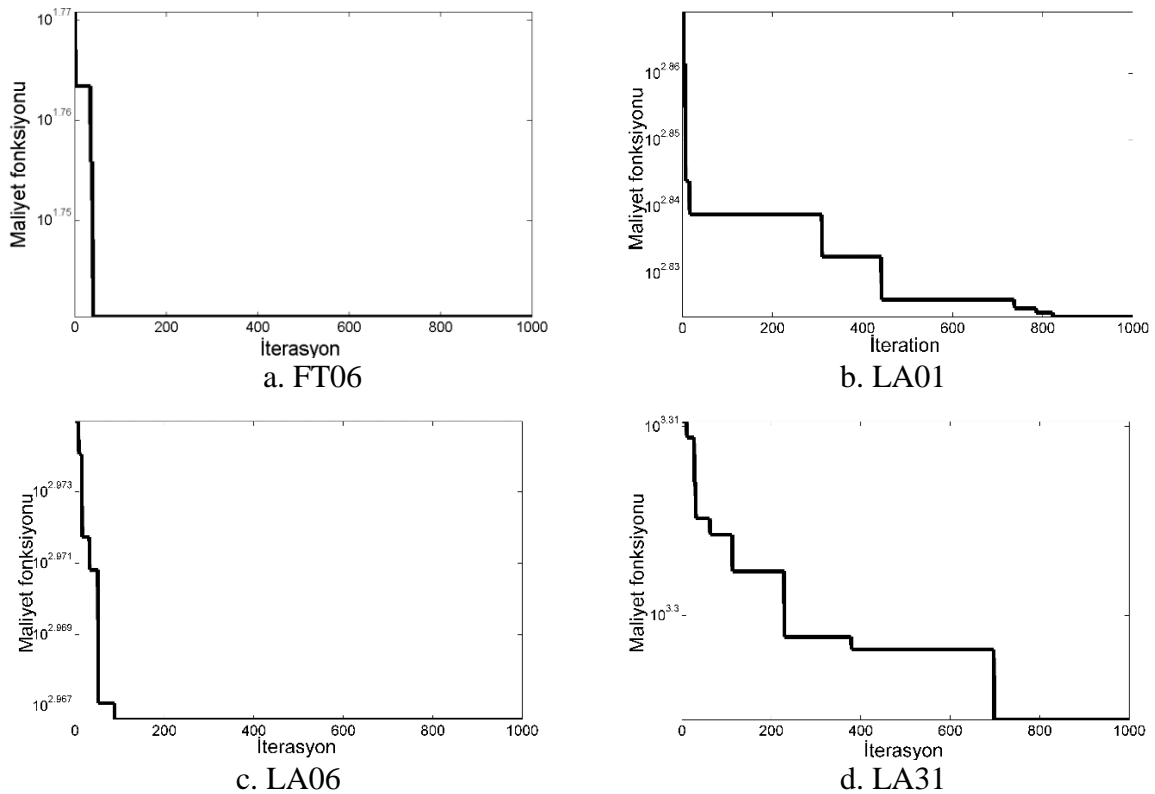
3.4. Operasyon (i, j) başlangıç zamanı olarak $e_{(i,j)}$ ata.

Adım 4: Eğer tamamlanmış çizelge elde edildiyse dur. Aksi durumda (i, j) operasyonunu Ω 'dan sil ve Ω 'da onu peşi sıra takip eden operasyonu ekle. Daha sonra 2. adıma git.

Giffler ve Thomson algoritmasıyla elde edilen aktif çizelgelerin maliyetleri her iterasyonda güncellenmekte ve kaydedilmektedir. Elde edilen maliyet değerleri literatürdeki test problemleriyle karşılaştırılmıştır. Bir sonraki bölümde deneysel sonuçlara yer verilecektir.

5. DENEYSEL SONUÇLAR

Algoritmanın etkililiğini göstermek amacıyla bir dizi atölye çizelgeleme problemi üzerinde deneyler yapılmış ve sonuçlar literatürle karşılaştırılmıştır. Test problemleri yöneylem araştırması kütüphanesinden, küçük, orta ve büyük ölçekli problemler arasından derlenmiştir (Beasley, 1990). Fisher ve Thomson tarafından düzenlenen FT06 ve Lawrence tarafından LA01, LA06 ve LA31 problemleri ele alınmış ve çözülmüştür (Fisher ve Thompson, 1963; Lawrence, 1984). Küçük ölçekli bir problem olan FT06 6 iş ve 6 makineden oluşurken, orta ölçekli problemler LA01 ve LA06 problemleri sırasıyla 10 iş ve 5 makine ile 15 iş ve 5 makineden oluşmaktadır. Büyük ölçekli bir problem olan LA31 problemi ise 30 iş ve 10 makineden oluşmaktadır. Bu test problemleri özellikle literatürde bu konuda çalışma yapan araştırmacıların hemen hemen hepsinin ortak olarak kullandığı test problemleri arasındadır. Her test problemi için populasyon sayısı 100, maksimum iterasyon sayısı 1000 olacak şekilde 10 replikasyon yapılmıştır. Maksimum, ortalama ve standart sapma değerleri verilmiştir. Şekil 5'te her bir test problemine ait yakınsama grafikleri verilmiştir.



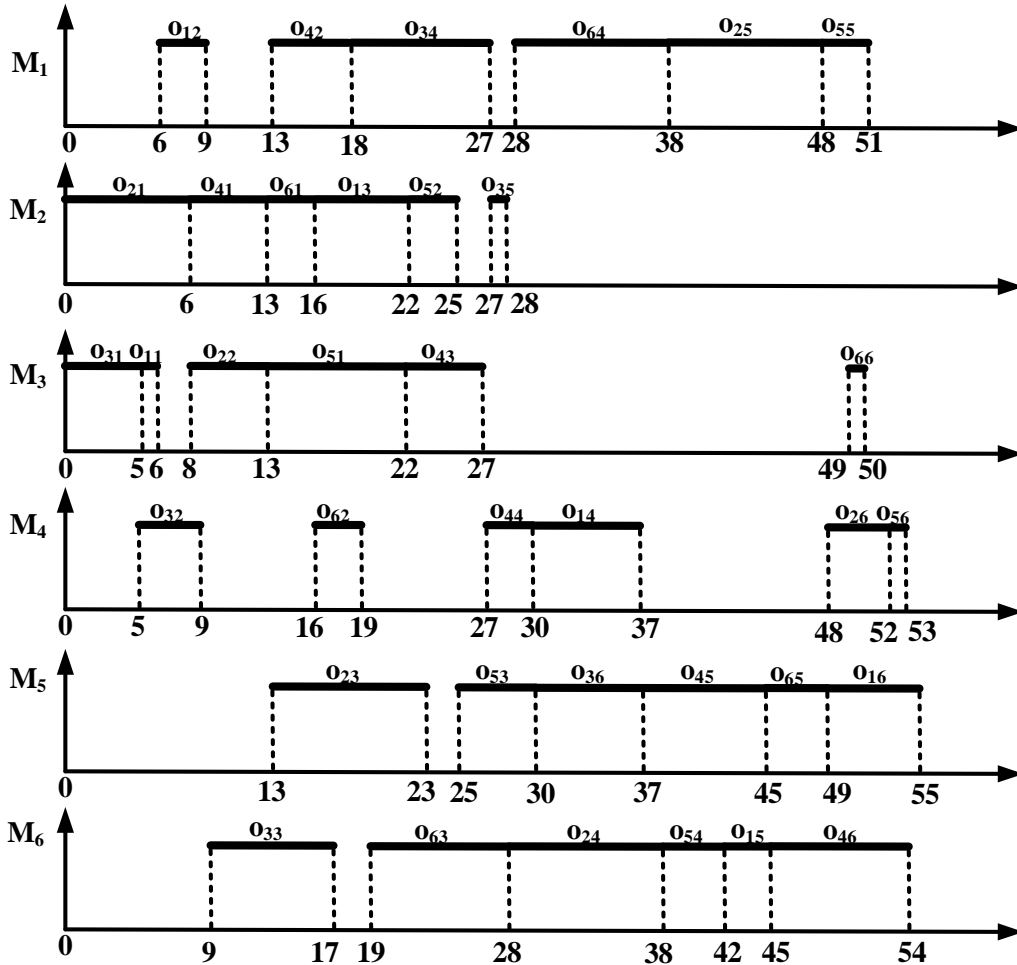
Şekil 5. Test problemlerine ait yakınsama grafikleri

Küçük ve orta ölçekli problemlerde algoritma oldukça hızlı bir biçimde en iyi çözüme yakınsarken, problem boyutu arttığında, yakınsama süresi uzamaktadır. Mevcut yöntem her bir test probleminde maksimum iterasyon sayısına erişmeden önce en iyi çözüme yakınsamayı başarmıştır. Çizelge 1’de her bir test problemi için deney sonuçları yer almaktadır.

Çizelge 1. Deney sonuçları

Problem #	Boyut ($J \times M$)	Optimum	Ortalama	Standart Sapma
FT06	6×6	55	57.4	1.05
LA01	10×5	666	667.3	8.16
LA06	15×5	926	931.6	7.63
LA31	30×10	1784	1995.3	28.34

Şekil 6’da FT06 probleminin optimum çözümünden elde edilen en iyi operasyon sırasını gösteren çizelge Gantt diyagramı yardımıyla gösterilmiştir. Örneğin başlangıç diliminde M_2 makinesine 2. iş atanmıştır. M_2 ’den 6. Zaman diliminde çıkan işin ardından aynı makineye 4. iş atanmıştır. Benzer okuma sürdürülürse, sistemden en son 1. işin 6. operasyonunun 55. zaman diliminde çıktığı görülecektir. Problemin en iyi çözümü literatürde de 55 olarak kaydedilmiştir.



Şekil 6. FT06 problemine ait Gantt çizelgesi

Çizelge 2’de ise mevcut yöntemin sonuçları Giffler ve Thomson algoritması benzer şekilde kullanan literatürdeki diğer çalışmalarla karşılaştırılmıştır.

Çizelge 2. Mevcut yöntemin diğer algoritmalarla karşılaştırılması

Algoritma	Kaynak	FT06	LA01	LA06	LA31
Parçacık Sürüsü Optimizasyonu	(Ivers ve Yen, 2007)	55	666	926	1784
Hibrit Akıllı Algoritma	(Hong-Wei vd., 2008)	55	666	926	1784
Memetik Algoritma	(Yang vd., 2008)	55	666	926	1784
Hibrit Parçacık Sürüsü	(Sha ve Hsu, 2006)	55	666	926	1784
Öğretme Öğrenme Algoritması	(Baykasoğlu vd., 2014)	55	666	926	1784
Kril Sürüsü Algoritması	-	55	666	926	1784

Çizelge 2’de görüldüğü gibi kril sürüsü algoritması standart test problemlerinde diğer algoritmalar gibi iyi bir performans göstererek literatürde bilinen en iyi çözümleri bulmayı başarmıştır. Mevcut araştırmanın öncül sonuçları neticesinde kril sürüsü algoritmasının çizelgeleme problemlerinin çözümünde iyi bir alternatif olma potansiyeline sahip olduğu söylenebilir.

6. SONUÇLAR

Bu çalışmada kril sürülerinin beslenme davranışlarından esinlenilerek literatüre kazandırılmış olan kril sürüsü algoritması ilk defa atölye tipi çizelgeleme problemine uygulanmıştır. Atölye tipi çizelgeleme problemi, tipik bir kombinatoriyal optimizasyon problemi olup, algoritmanın diğer kombinatorik problemlerdeki performansı hakkında ön bilgi sağlamaktadır. Kril sürüsü algoritması incelenen test problemlerinde iyi sonuçlar vermiştir. Gelecek çalışmalarda literatürdeki farklı kombinatoriyal problemler çözülerek daha ayrıntılı ve genel bir karşılaştırma yapılacaktır. Ayrıca algoritma farklı yerel arama metotlarıyla birleştirilerek farklı hibrit yaklaşımlar önerilmesi üzerinde durulacaktır.

KAYNAKLAR

- Adams J., Balas E., Zawack D. (1988): "The Shifting Bottleneck Procedure for Job Shop Scheduling", Management Science, Cilt 34, No. 3, s.391-401.
- Aiex R. M., Binato S., Resende M. G. C. (2003): "Parallel GRASP with Path-Relinking for Job Shop Scheduling", Parallel Computing, Cilt 29, No. 4, s.393-430.
- Anandaraman C. (2011): "An Improved Sheep Flock Heredity Algorithm for Job Shop Scheduling and Flow Shop Scheduling Problems", International Journal of Industrial Engineering Computations, Cilt 2, No. 4, s749-764.
- Baker K. R. (1974): "Introduction to Sequencing and Scheduling", New York: Wiley.
- Baykasoğlu A. (2002): "Linguistic-Based Meta-Heuristic Optimization Model for Flexible Job Shop Scheduling", International Journal of Production Research, Cilt 40, No. 17, 4523-4543.
- Baykasoğlu A., Hamzadayı A., Köse S. Y. (2014): "Testing the Performance of Teaching–Learning Based Optimization (TLBO) Algorithm on Combinatorial Problems: Flow Shop and Job Shop Scheduling Cases", Information Sciences, Cilt 276, No. 0, s.204-218.
- Beasley J. E. (1990): "OR-Library: Distributing Test Problems by Electronic Mail", Journal of the Operational Research Society, Cilt 41, No. 11, s.1069-1072.
- Bierwirth C., Mattfeld D. C. (1999): "Production Scheduling and Rescheduling with Genetic Algorithms", Evolutionary Computation, Cilt 7, No. 1, s.1-17.
- Brucker P., Jurisch B., Sievers B. (1994): "A Branch and Bound Algorithm for the Job-Shop Scheduling Problem", Discrete Applied Mathematics, Cilt 49, No. 3, s.107-127.

- Cheng R., Gen M., Tsujimura Y. (1996): "A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms—I. Representation", *Computers and Industrial Engineering*, Cilt 30, No. 4, s.983-997.
- Fisher H., Thompson G. L. (1963): "Probabilistic Learning Combinations of Local Job Shop Scheduling Rules", Englewood Cliffs, NJ: Prentice-Hall.
- Gandomi A. H., Alavi A. H. (2012): "Krill Herd: A New Bio-Inspired Optimization Algorithm", *Communications in Nonlinear Science and Numerical Simulation*, Cilt 17, No. 12, s.4831-4845.
- Gen M., Cheng R. (2000): "Genetic Algorithms and Engineering Optimization", Cilt 7, John Wiley & Sons.
- Giffler B., Thompson G. L. (1960): "Algorithms for Solving Production-Scheduling Problems", *Operations Research*, Cilt 8, No. 4, s.487-503.
- Gonçalves J. F., Resende M. G. C. (2014): "An Extended Akers Graphical Method with a Biased Random-Key Genetic Algorithm for Job-Shop Scheduling", *International Transactions in Operational Research*, Cilt 21, No. 2, s.215-246.
- Hasan S. M. K., Sarker R., Essam, D., Cornforth D. (2009): "Memetic Algorithms for solving job-shop scheduling problems", *Memetic Computing*, Cilt 1, No. 1, 69-83.
- Hofmann E. E., Haskell A. G. E., Klinck J. M., Lascara C. M. (2004): "Lagrangian Modelling Studies of Antarctic Krill (*Euphausia Superba*) Swarm Formation", *ICES Journal of Marine Science*, Cilt 61, No. 4, s.617-631.
- Hong Wei G., Liang S., Yan Chun L., Feng Q. (2008): "An Effective PSO and AIS-Based Hybrid Intelligent Algorithm for Job-Shop Scheduling", *Systems, Man and Cybernetics, Part A: Systems and Humans*, Cilt 38, No. 2, s.358-368.
- Jain A. S., Meeran S. (1999): "Deterministic Job-Shop Scheduling: Past, Present and Future", *European Journal of Operational Research*, Cilt 113, No. 2, s.390-434.
- Lawrence S. (1984): "Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplement)", Pittsburg: Carnegie Mellon University.
- Lin T. L., Horng S. J., Kao T. W., Chen Y. H., Run R. S., Chen R. J., Kuo I. H. (2010): "An Efficient Job-Shop Scheduling Algorithm Based on Particle Swarm Optimization", *Expert Systems with Applications*, Cilt 37, No. 3, s.2629-2636.
- Luh G. C., Chueh C. H. (2009): "A Multi-Modal Immune Algorithm for the Job-Shop Scheduling Problem", *Information Sciences*, Cilt 179, No. 10, s.1516-1532.
- Nowicki E., Smutnicki C. (1996): "A Fast Taboo Search Algorithm for the Job Shop Problem", *Management Science*, Cilt 42, No. 6, 797-813.
- Potts C. N., Strusevich V. A. (2009): "Fifty Years of Scheduling: a Survey of Milestones", *Journal of the Operational Research Society*, Cilt 60, No. 1, s.41-s.68.
- Price H. J. (1989): "Swimming Behavior of Krill in Response to Algal Patches: A Mesocosm Study", *Limnology and Oceanography*, Cilt 34, No. 4, s.649-659.
- Roy P. K., Paul C. (2014): "Optimal Power Flow Using Krill Herd Algorithm", *International Transactions on Electrical Energy Systems*.
- Sha D. Y., Hsu C. Y. (2006): "A Hybrid Particle Swarm Optimization for Job Shop Scheduling Problem", *Computers and Industrial Engineering*, Cilt 51, No. 4, s.791-808.
- Singh V., Sood M. M. (2013): "Krill Herd Clustering Algorithm Using DBSCAN Technique", *International Journal of Computer Science & Engineering Technology*, Cilt 4, No. 3. s.197-201
- Wang G. G., Gandomi A. H., Alavi A. H. (2013): "A Chaotic Particle-Swarm Krill Herd Algorithm for Global Numerical Optimization", *Kybernetes*, Cilt 42, No. 6, s.962-978.

- Wang G. G., Gandomi A. H., Alavi A. H. (2014): "An Effective Krill Herd Algorithm with Migration Operator in Biogeography-Based Optimization", Applied Mathematical Modelling, Cilt 38, No. 9, s.2454-2462.
- Wang G. G., Gandomi A. H., Alavi A. H. (2014): "Stud Krill Herd Algorithm", Neurocomputing, Cilt 128, No. 1, s.363-370.
- Wang G. G., Guo L., Gandomi A. H., Alavi A. H., Duan H. (2013): "Simulated Annealing-Based Krill Herd Algorithm for Global Optimization", Abstract and Applied Analysis, Cilt 11.
- Wang G., Guo L., Gandomi A. H., Cao L., Alavi A. H., Duan H., Li J. (2013): "Lévy-Flight Krill Herd Algorithm", Mathematical Problems in Engineering, Cilt 2013, Article ID 682073, doi:10.1155/2013/682073
- Wang G., Guo L., Wang H., Duan H., Liu L., Li J. (2014): "Incorporating Mutation Scheme into Krill Herd Algorithm for Global Numerical Optimization", Neural Computing and Applications, Cilt 24, No. 3, s.853-871.
- Yang J. H., Sun L., Lee H. P., Qian Y., Liang Y. C. (2008): "Clonal Selection Based Memetic Algorithm for Job Shop Scheduling Problems", Journal of Bionic Engineering, Cilt 5, No. 2, s.111-119.