

2018 Vol:5 / Issue:24	pp.1815- 1821	Article Arrival Date (Makale Geliş Tarihi) The Published Rel. Date (Makale Yayın Kabul Tarihi) The Published Date (Yayınlanma Tarihi)	01.08.2018 27.09.2018 27.09.2018
--------------------------	------------------	---	--

## BLOKLAMA KISITLI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN MAYMUN ARAMA ALGORİTMASI İLE ÇÖZÜMÜ

### SOLUTION OF BLOCKING RESTRICTION FLOW SHOP SCHEDULING PROBLEMS WITH MONKEY SEARCH ALGORITHM

İskender SAÇ

Konya Teknik Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi Endüstri Mühendisliği  
Bölümü, Konya / TÜRKİYE, ORCID: 0000-0002-2504-7935

Prof. Dr. Orhan ENGİN

Konya Teknik Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi Endüstri Mühendisliği  
Bölümü, orhanengin@yahoo.com, Konya / TÜRKİYE, ORCID: 000-0002-7250-0317

#### ÖZET

Akış tipi çizelgeleme problemlerinde  $m$ -makine ve  $n$ -iş bulunmaktadır. Bütün işler sırasıyla bütün makinelerde işlem görür. Bloklama kısıtı, bir sonraki makine dolu olduğunda, işlerin operasyona tabi tutulduğu makineden ayrılamama durumunu ifade eder. Bu çalışmada, bloklama kısıtlı akış tipi çizelgeleme problemlerinin çözümü amaçlanmıştır. Bloklama kısıtlı akış tipi çizelgeleme probleminde, toplam akış zamanını minimize etmek için maymun arama algoritması geliştirilmiştir. Maymun arama algoritması, maymunların turmanış sürecinin simülasyonuna dayanan meta-sezgisel algoritmalarından biridir. Araştırmada, bloklama kısıtlı akış tipi çizelgeleme problemleri ve çözüm aşamasında kullanılan algoritmalar tanımlanmıştır. Literatürde yer alan, bloklama kısıtlı akış tipi çizelgeleme problemleri, maymun arama algoritması ile çözülmüştür. Elde edilen sonuçlar, literatürdeki çözümler ile karşılaştırılmıştır. Maymun arama algoritmasının, bloklama kısıtlı akış tipi çizelgeleme problemlerinin çözümünde başarılı sonuçlar verdiği belirlenmiştir.

**Anahtar Kelimeler:** Bloklama kısıtlı akış tipi çizelgeleme, Maymun arama algoritması, NP-Zor çizelgeleme

#### ABSTRACT

There are  $m$ -machines and  $n$ -jobs in flow shop scheduling problems. All jobs are processed on all machines respectively. Blocking constraint refers to the inability to leave the machine from which the work is subjected to operation when the next machine is full. In this study, it is aimed to solve the problems of blocking restricted flow type scheduling. In the blockage constrained flow type scheduling problem, the monkey search algorithm has been developed to minimize the total flow time. The monkey search algorithm is one of the meta-heuristic algorithms based on the simulation of monkeys' climbing process. In the study, blocking limited flow type scheduling problems and the algorithms used in the solution phase have been defined. Blocking constrained flow type scheduling problems in the literature have been solved by the monkey search algorithm. The results were compared with the solutions in the literature. Monkey search algorithm has been found to be successful in solving blocking limited flow type scheduling problems.

**Keywords:** Blocking restriction flow shop scheduling, Monkey search algorithm, NP-Hard scheduling

#### 1. GİRİŞ

Çizelgeleme, birçok imalat ve servis endüstrisinde önemli rol oynayan bir karar verme prosesidir. Akış tipi çizelgeleme problemlerinde, birbirinden farklı  $m$ -makine ve  $n$ -iş mevcut olup her bir iş aynı sıra ile

$m$  farklı operasyondan oluşur (Fıglalı ve Engin 2002: 1). Akış tipi çizelgeleme problemleri ile ilgili ilk çalışmayı 2-makine  $n$ -iş ile Johnson S.M. (1954) yapmıştır. Bloklamalı akış tipi problemlerinde, akış tipi çizelgeleme problemlerinden farklı olarak bir makinede işlenen iş bittikten sonra bir sonraki makine bir işe sahip ise işlenmesi biten iş, bir sonraki makine boşalana kadar son işlendiği makineyi terk edemez. Graham ve diğerleri (1979), bloklamalı akış tipini,  $F_m/block/C_{max}$  notasyonu ile tanımlamışlardır. Hall ve Sriskandarajah (1996), akış atölyesinde bloklama ve artık süreç (boşta bekleme süresi) ile ilgili araştırma yapmıştır. Çalışmalarında, bloklamalı ve boşta bekleme süresi bulunan bazı problemlerin karmaşıklığını açıklamışlar ve bunların çözümü için bazı sezgisel yöntemler önermişlerdir. Grabowski ve Pempera (2007), ikiden fazla makineden oluşan bloklamalı akış tipi çizelgeleme problemlerinin NP-hard olduğunu göstermişlerdir. Bu sistemlerin karmaşıklığını çözmek için sezgisel yaklaşımlar kullanılmıştır. Wang ve diğerleri (2010), bloklamalı akış tipi çizelgeleme problemlerini çözmek için hibrit harmoni arama algoritması geliştirmişlerdir. Davendra ve Bialic-Davendra (2013), bloklamalı akış tipi çizelgeleme problemlerini çözmek için kendini organize eden göçmen algoritması sunmuşlardır ve Taillard problemlerinin çözümünde üst sınır elde etmişlerdir. Ribas ve Companys (2015), bloklamalı akış tipi çizelgeleme problemlerinin toplam akış süresi minimizasyonu için etkili sezgisel yöntemler önermişlerdir. Çalışmalarında, ikiden fazla makineye sahip bloklamalı akış tipi problemlerini maymun arama algoritması ile çözmüşlerdir. Ribas ve diğerleri (2017), paralel akış tipinde bloklamalı çizelgeleme problemlerini çözmek için etkili sezgisel yöntemler geliştirmişlerdir. Maymun arama algoritması, maymunların dağa tırmanış sürecinin simülasyonuna dayanan meta-sezgisel algoritmalarından biridir. Maymun arama algoritması 2008’ de Zhao ve Tang (2008) tarafından geliştirilen bir algoritmadır. Bu algoritma, Tao ve diğerleri (2010) tarafından proje çizelgeleme problemlerini çözmek için kullanılmıştır. Daha sonra, Wang ve diğerleri (2010) iletim şebekesi problemlerini çözmek için kullanmıştır. Maymun arama algoritması, Zhang ve diğerleri (2011) tarafından saldırı tespiti ve Yi ve diğerleri (2012) tarafından yapısal sağlık izleme problemlerini çözmek için kullanılmıştır. Marichelvam ve diğerleri (2017) akış tipi çizelgeleme problemlerini çözmek için bir hibrit maymun arama algoritması geliştirmişlerdir.

## 2. BLOKLAMALI AKIŞ TİPİ ÇİZELGELEME

Bloklamalı akış tipi çizelgeleme problemlerinde bütün işler sırasıyla, makine 1’den makine  $m$ ’ye kadar işlenmelidir. Yükleme süreleri işlem sürelerine dahildir. Bir iş makine 1’de işlendikten sonra 2. makineye geçer, daha sonra 2. makinede işlenir ve 3. makineye geçer. Bu şekilde  $m$ . makineye kadar devam eder. Bir makinede iş işlendikten sonra eğer bir sonraki makine boş değilse son işlendiği makinede beklemek zorundadır. Bir sonraki makine boşaldığında, iş bir sonraki makineye geçer. Bu beklemeden kaynaklanan fazla zamana bloklama süresi denir. Bir makinedeki iş, bir sonraki makineye geçtikten sonra o makineye bir önceki makineden bir sonraki iş gelene kadar makine boş bekler. Bu zamana ise boşta bekleme zamanı denir. Bu süreler Şekil 1’de gösterilmektedir. Bloklamalı akış tipi çizelgeleme problemlerinin formülasyonu aşağıda verilmiştir (Han vd., 2016:6784). Formülasyonda kullanılan terimler şunlardır.

$b_{i,j} \rightarrow$  i. makinedeki, j. işin başlama zamanı

$c_{i,j} \rightarrow$  i. makinedeki, j. işin bitiş zamanı

$u_{i,j} \rightarrow$  i. makinedeki, j. işin işleme süresi

$C_{max}(S) \rightarrow$  S sıralamasının  $C_{max}$  değeri

$$b_{0,1} = 0; \quad (1)$$

$$c_{i,1} = b_{i-1,1} + u_{i,1} \quad i = 1, 2, \dots, m-1 \quad (2)$$

$$b_{0,j} = c_{1,j-1} \quad j = 2, 3, \dots, n \quad (3)$$

$$c_{i,j} = \max(c_{i-1,j} + u_{i,j}, c_{i+1,j-1}) \quad i = 1, 2, \dots, m-1 \quad j = 2, 3, \dots, n \quad (4)$$

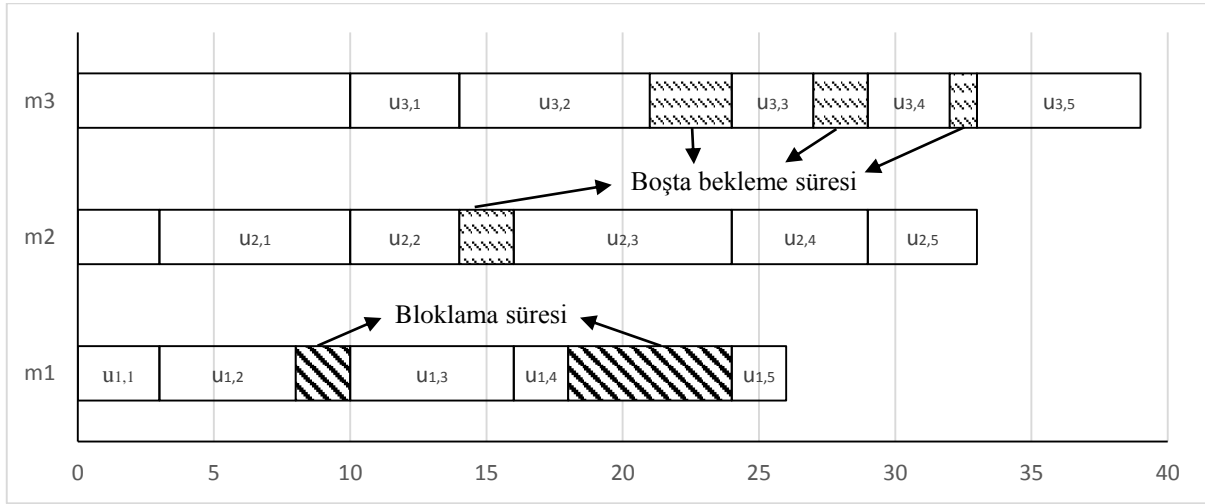
$$c_{m,j} = c_{m-1,j} + u_{m,j} \quad j = 1, 2, \dots, n \quad (5)$$

$$C_{max}(S) = c_{m,n} \quad (6)$$

(1) Birinci işin, ilk makinede işe başlama süresidir. (2) Birinci işin 1' den  $m-1$ . makineye kadar işlerin bitiş zamanını ifade eder. (3) İkinci işten  $n$ . işe kadar işlerin başlama süresinin hesaplanmasını gösterir. (4) 1'den  $m-1$ . makineye ve 2'den  $n$ . işe kadar olan bütün işlerin bitiş zamanını ifade eder. (5)  $m$ . makinedeki bütün işlerin bitiş zamanının hesaplanmasında kullanılır. (6) Bütün işlerin bitiş zamanını gösterir.

*Örnek* Bloklamalı akış tipi çizelgeleme problemi, 3-makine x 5-iş'lik bir örnek üzerinde aşağıda gösterilmiştir. Beş işin, üç makinedeki işlem süreleri (7) numaralı matris ile verilmiştir.

$$u_{3 \times 5} = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} & u_{1,5} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} & u_{2,5} \\ u_{3,1} & u_{3,2} & u_{3,3} & u_{3,4} & u_{3,5} \end{bmatrix} = \begin{bmatrix} 3 & 5 & 6 & 2 & 2 \\ 7 & 4 & 8 & 5 & 4 \\ 4 & 7 & 3 & 3 & 6 \end{bmatrix} \quad (7)$$



Şekil 1. Gantt Şeması Örneği

Yukarıda verilen örnekleme,  $S = \{1,2,3,4,5\}$  sıralaması için aşağıdaki (8) nolu ifade yardımı ile tamamlanma zamanı hesaplanmıştır.

$$\begin{aligned} b_{0,1} &= 0 \\ c_{1,1} &= b_{0,1} + u_{1,1} = 3; \quad c_{2,1} = c_{1,1} + u_{2,1} = 10; \quad c_{3,1} = c_{2,1} + u_{3,1} = 14; \\ c_{0,2} &= c_{1,1} = 3; \\ c_{1,2} &= \max\{c_{0,2} + u_{1,2}, c_{2,1}\} = 10; \\ c_{2,2} &= \max\{c_{1,2} + u_{2,2}, c_{3,1}\} = 14; \\ c_{3,2} &= c_{2,2} + u_{3,2} = 21; \\ c_{0,3} &= c_{1,2} = 10; \\ c_{1,3} &= \max\{c_{0,3} + u_{1,3}, c_{2,2}\} = 16; \\ c_{2,3} &= \max\{c_{1,3} + u_{2,3}, c_{3,2}\} = 24; \\ c_{3,3} &= c_{2,3} + u_{3,3} = 27; \\ c_{0,4} &= c_{1,3} = 16; \\ c_{1,4} &= \max\{c_{0,4} + u_{1,4}, c_{2,3}\} = 24; \\ c_{2,4} &= \max\{c_{1,4} + u_{2,4}, c_{3,3}\} = 29; \end{aligned} \quad (8)$$

$$c_{3,4} = c_{2,4} + u_{3,4} = 32;$$

$$c_{0,5} = c_{1,4} = 24;$$

$$c_{1,5} = \max\{c_{0,5} + u_{1,5}, c_{2,4}\} = 29;$$

$$c_{2,5} = \max\{c_{1,5} + u_{2,5}, c_{3,4}\} = 33;$$

$$c_{3,5} = c_{3,4} + u_{3,5} = 39;$$

$$C_{max}(S) = c_{m,n} \rightarrow C_{max}(1,2,3,4,5) = c_{3,5} = 39;$$

Yukarıdaki grafikte ve yapılan işlemler sonucunda görüldüğü gibi verilen örneklemin  $S=\{1,2,3,4,5\}$  sıralaması için min  $C_{max}$  değeri 39 olarak hesaplanmış ve Şekil 1'de Gantt diyagramı üzerinde gösterilmiştir.

### 3. MAYMUN ARAMA ALGORİTMASI

Maymun arama algoritması, maymunların dağa tırmanış sürecinin simülasyonuna dayanan bir meta-sezgisel algoritmadır. Maymun arama algoritmasında ilk olarak maymunların nüfus büyüklüğü tanımlanır. Ardından her maymunun konumu  $x_i$  vektörü ile gösterilir. Bu pozisyon rastgele oluşturulur. Maymunların konumu adım adım tırmanma işlemi ile değiştirilir. Tırmanış süreci, amaç fonksiyonunun değerini iyileştirmek için kullanılır. Tırmanma işleminden sonra her maymun kendi tepesine ulaşır. Ardından, her maymun şimdiki konumundan daha yüksek puan arar. Daha yüksek bir dağ bulunursa, maymun oraya görme yeteneğini kullanarak atlayacaktır. Bir maymunun görme gücü, maymunun izleyebileceği maksimum mesafedir. Maymunların pozisyonu güncelleştirilir. Ardından, mevcut pozisyonlar pivot olarak kullanılarak yeni alanlar bulunur. Bu aşama takla atma süreci olarak bilinir ve maymunlar için yeni bir pozisyon verir. Amaç fonksiyonu değerleri incelenir. Yineleme sayısı tamamlanırsa süreç durdurulur. Algoritmanın ana süreçleri aşağıdaki gibidir (Marichelvam vd., 2017:84).

(1) Maymunların nüfus büyüklüğü ( $M$ ) tanımlanır, her maymun  $i \in \{1, 2, \dots, M\}$  şeklinde tanımlanır.  $n$  boyutlu bir problemde maymunların ilk pozisyonları  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$  şeklindedir.

(2) Başla;

Maymunların başlangıç çözümleri rasgele üretilir.

(3) Tırmanma Süreci;

I.  $\Delta x_i = (\Delta x_{i1}, \Delta x_{i2}, \dots, \Delta x_{in})$  Şeklinde bir rasgele vektör üretilir.  $\Delta x_{i,j}, \alpha$  olarak ayarlanır.  $\alpha$ 'ya tırmanış sürecinin basamak uzunluğu denir.

$$II. f'_{ij}(x_i) = \frac{f(x_i + \Delta x_i) - f(x_i - \Delta x_i)}{2\Delta x_{ij}}, \quad j = 1, 2, \dots, n$$

$x_i$  vektörü, amaç fonksiyonunun sözde gradyanı olarak adlandırılır.

$$III. y_j = x_{ij} + \text{asign}(f'_{ij}(x_i)), \quad j = 1, 2, \dots, n$$

$$y = (y_1, y_2, \dots, y_n)$$

IV. Eğer  $y$  uygulanabilir ise  $x_i \leftarrow y$

değilse  $x_i$  aynı kalır.

V. İlgili iterasyonda, amaç fonksiyonda çok az değişiklik olana kadar ya da  $N_c$ 'nin ulaşabileceği maksimum iterasyon sayısına kadar, 1-4 arasındaki adımlar tekrarlanır.

## (4) İzleme – Atlama Süreci;

- I.  $(x_{ij} - b, x_{ij} + b)$ ,  $j = 1, 2, \dots, n$  fonksiyonu ile gerçek  $y_i$  sayısı rasgele üretilir.  $b$ ' ye maymunun maksimum görme mesafesi denir ve maymunun izleyebileceği maksimum mesafeyi belirtir.
- II.  $f(y) > f(x_i)$  ise ve  $y$  uygulanabilir ise  $x_i \leftarrow y$   
değilse,  $y$  uygun bir nokta bulana kadar veya belirli bir sayıda izleme süresine ulaşana kadar 1.adım tekrarlanır.
- III.  $y$  başlangıç pozisyonu olarak kullanılarak tırmanış süreci tekrarlanır.

## (5) Takla Süreci;

- I.  $[c, d]$  aralığından rasgele bir gerçek  $\theta$  sayısı üretilir.
- II.  $p_j = \frac{1}{M} \sum_{i=1}^M x_{ij}$ ,  $j = 1, 2, \dots, n$   
 $y_j = x_{ij} + \theta(p_j - x_{ij})$   
 $p = (p_1, p_2, \dots, p_n)$  takla atımı pivotu,  
 $(p_j - x_{ij})$  maymun  $i$ ' nin takla atımı yönüdür.
- III. Eğer  $y = (y_1, y_2, \dots, y_n)$  uygulanabilir ise  $x_i \leftarrow y$   
değilse uygun  $y$  bulunana kadar 1. ve 2. adım tekrarlanır.

## (6) Sonlandır.

**4. DENEYSSEL DEĞERLENDİRME**

Bu araştırmada, Carlier (1978) tarafından geliştirilen akış tipi çizelgeleme problemleri, bloklamalı akış tipi çizelgeleme olarak maymun arama algoritması ile çözülmüştür. Carlier' in 8 adet örneği bulunmaktadır. Bunlar; Car1 (11 iş x 5 makine), Car2 (13x4), Car3 (12x5), Car4(14x4), Car5(10x6), Car6(8x9), Car7(7x7) ve Car8(8x8) dir. Bloklamalı akış tipi çizelgeleme problemlerinin maymun arama algoritması ile çözümünü C# yazılım dilinde, Visual Studio 2017 programında, Intel (R) Core (TM) i5-4210U CPU 1.70 GHz 2.40 GHz işlemcili, 4 GB Ram ve Windows 10 Home işletim sistemi bulunan bir bilgisayarda hazırlanan program yardımıyla çözülmüştür. Elde edilen sonuçlar Tablo 1'de sunulmuştur.

**Tablo 1.** Carlier (1978) Bloklamalı Akış Tipi Problemlerinin  $c_{max}$  Değerleri

Problemler	En iyi $C_{max}$ Değerleri	MAA ile hesaplanan $C_{max}$ Değerleri	Hesaplama Süreleri (sn)
Car1	7409	7409	0.96
Car2	7503	7503	0.83
Car3	8014	8014	0.91
Car4	8738	8738	0.93
Car5	8218	8218	0.84
Car6	8873	8873	0.96
Car7	6788	6788	0.66
Car8	8585	8585	0.84

Tüm problemler için başlangıç çözümleri NEH algoritması (Nawaz vd., 1983) kullanılarak oluşturulmuştur. Oluşturulan bu sıralamaya, sırayla tırmanma, atlama ve takla işlemleri uygulanarak yeni çözüm aranmıştır. Bu işlemlerin sonucunda ortaya çıkan değerler, Carlier' in bloklamalı problemlerinin en iyi değerler olan Davendra ve Bialic-Davendra da (2013:2214) verilen sonuçlarla karşılaştırılmıştır.

Tablo 1' de görüldüğü gibi Carlier' in (1978) sekiz akış tipi çizelgeleme problemi, bloklama kısıtı altında önerilen maymun arama algoritması ile çözülerek en iyi  $C_{max}$  değerlerine ulaşılmıştır.

## 5. SONUÇ

Akış tipi çizelgeleme problemlerinde, işler sırasıyla bütün makinelerde işlem görür. Bloklama kısıtlı akış tipi çizelgeleme problemlerinde ise bir sonraki makine dolu olduğunda, işler operasyona tabi tutulduğu makinaneden ayrılamazlar. Bu çalışmada, NP-zor olarak bilinen bloklamalı akış tipi çizelgeleme problemleri, yeni önerilen maymun arama algoritması ile çözülmüştür. Elde edilen sonuçlar literatürdeki değerler ile karşılaştırılmıştır. Maymun arama algoritmasının bloklamalı akış tipi çizelgeleme problemlerinin çözümünde başarılı sonuçlar verdiği belirlenmiştir.

## KAYNAKÇA

- CARLIER J., 1978, Ordonnancements a Contraintes Disjonctives, *R.A.I.R.O. Recherche Operationnelle/Operations Research* 12, 333-351.
- DAVENDRA D. ve BIALIC-DAVENDRA M., 2013, Scheduling Flow Shops With Blocking Using a Discrete Self-Organising Migrating Algorithm, *International Journal of Production Research*, 51(8), 2200-2218.
- FIĞLALI A. ve ENGİN O., 2002, Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma (GA) ile Çözüm Performansının Artırılmasında Deneysel Tasarımı Uygulaması, *Endüstri Mühendisliği Dergisi*, 13(3), 2 – 7.
- GRABOWSKİ J. ve PEMPERA J., 2007, The Permutation Flow Shop Problem With Blocking, A Tabu Search Approach, *Omega*, 35(3), 302-311.
- GRAHAM R. L., LAWLER E. L., LENSTRA J. K., ve KAN A.R., 1979, Optimization and Approximation in Deterministic Sequence and Scheduling: A Survey, *Annals of Discrete Mathematics*, 5, 287-326.
- HALL N. G. ve SRISKANDARAJAH C., 1996, A Survey of Machine Scheduling Problems With Blocking and No-Wait in Process, *Operations Research*, 44(3), 510-525.
- HAN Y., GONG D., LI J. ve ZHANG Y., 2016, Solving the Blocking Flow Shop Scheduling Problem With Makespan Using a Modified Fruit Fly Optimisation Algorithm, *International Journal of Production Research*, 54(22), 6782-6797.
- JOHNSON S.M., 1954, Optimal Two- and Three- Stage Production Schedules With Setup Times Included, *Nav. Res. Logist. Q.*, 1(1), 61-68.
- MARICHELVA M.K., TOSUN Ö. ve GEETHA M., 2017, Hybrid Monkey Search Algorithm for Flow Shop Scheduling Problem Under Makespan and Total Flow Time, *Applied Soft Computing*, 55, 82-92.
- NAWAZ M., ENSCORE E. E., ve HAM I., 1983, A Heuristic Algorithm for The M-Machine, N-Job Flow-Shop Scheduling Problem, *Omega*, 11(1), 91-95.
- RİBAS I. ve COMPANYS R., 2015, Efficient Heuristic Algorithms for the Blocking Flow Shop Scheduling Problem With Total Flow Time Minimization, *Computers & Engineering*, 87, 30-39.
- RİBAS I., COMPANYS R. ve TORT-MARTORELL X., 2017, Efficient Heuristics for the Parallel Blocking Flow Shop Scheduling Problem, *Expert Systems With Applications*, 74, 41-54.
- TAO Z., YU X., ve MALI Z., 2010, Optimization of Gas Filling Station Project Scheduling Problem Based on Monkey Algorithm, *Value Engineering*, 8(1), 90-92.

- YI T. H., LI H. N. ve ZHANG X. D., 2012, A Modified Monkey Algorithm for Optimal Sensor Placement in Structural Health Monitoring, *Smart Mater. Struct.*, 21(10), 65-69.
- WANG J. R., YU Y. X. ve ZENG Y., 2010, Discrete Monkey Algorithm and Its Application in Transmission Network Expansion Planning, *J. Tianjin Univ.*, 43(9), 798-803.
- WANG L., PAN Q, ve TASGETİREN M.F., 2010, Minimizing the Total Flow Time in a Flow Shop With Blocking by Using Hybrid Harmony Search Algorithms, *Expert Systems With Applications*, 37(12), 7929-7936.
- ZHANG J. J., ZHANG Y. P. ve SUN J. Z., 2011, Intrusion Detection Technology Based on Monkey Algorithm, *Computer Engineering*, 37(14), 131-133.
- ZHAO R. ve TANG W., 2008, Monkey Algorithm for Global Numerical Optimization, *Journal of Uncertain Systems*, 2(3), 165-176.