

## Atölye tipi çizelgeleme problemleri için parçacık sürü optimizasyonu yöntemi

Mehmet ŞEVKLİ\*, M. Mutlu YENİSEY

İTÜ İşletme Fakültesi, Endüstri Mühendisliği Bölümü, 34357, Maçka, İstanbul

### Özet

Popülasyon temelli sezgisel yöntemlerden biri olan Parçacık Sürü Optimizasyonu (PSO), kuş ve balık sürülerinin sosyal davranışlarından etkilenerek geliştirilen yeni bir eniyileme yöntemidir. Bu makalede, zor çizelgeleme problemleri arasında yer alan Atölye Tipi Çizelgeleme problemlerinin çözümü için, bir PSO modeli, Değişken Komşuluk Arama yöntemi ile birlikte geliştirilmiştir. Oluşturulan bu model, tamamlanma zamanı performans ölçütüne göre literatürde yer alan bazı zor test problemleri üzerindeki sonuçları incelenmiş ve iyi sonuçlar veren diğer sezgisel yöntemlerin sonuçlarıyla karşılaştırılmıştır. Sonuçta genel olarak önerilen modelin diğer yöntemlere göre daha iyi veya eşdeğer seviyede olduğu görülmüştür.

**Anahtar Kelimeler:** Atölye tipi çizelgeleme, parçacık sürü optimizasyonu, sezgiseller.

### A particle swarm optimization for the job shop scheduling problems

#### Abstract

Particle Swarm Optimization (PSO) is one of the population based optimization technique inspired by social behavior of bird flocking and fish schooling. PSO inventors were inspired of such natural process based scenarios to solve the optimization problems. In PSO, each single solution, called a particle, is considered as a bird, the group becomes a swarm (population) and the search space is the area to explore. Each particle has a fitness value calculated by a fitness function, and a velocity of flying towards the optimum, food. All particles fly across the problem space following the particle nearest to the optimum. PSO starts with initial population of solutions, which is updated iteration-by-iteration. Therefore, PSO can be counted as an evolutionary algorithm besides being a metaheuristics method, which allows exploiting the searching experience of a single particle as well as the best of the whole swarm. In this paper, A PSO model for the job shop scheduling problem is proposed. In addition, a simple but efficient local search method called Variable Neighborhood Search (VNS) is embedded to the PSO model and applied to several hardest benchmark suites. The results for the PSO algorithm with VNS are also presented and compared with many efficient meta-heuristic algorithms in literature. As a final result, PSO with VNS results are generally found to be better than other results.

**Keywords:** Job shop scheduling, particle swarm optimization, Meta-Heuristics.

---

\*Yazışmaların yapılacağı yazar: Mehmet ŞEVKLİ. msevкли@fatih.edu.tr; Tel: (212) 889 08 10 dahili: 1060.

Bu makale, birinci yazar tarafından İTÜ İşletme Fakültesi'nde tamamlanmış olan "atölye tipi çizelgeleme problemlerine parçacık sürü optimizasyonu yaklaşımı ve genetik algoritma modeli ile karşılaştırılması" adlı doktora tezinden hazırlanmıştır. Makale metni 08.06.2005 tarihinde dergiye ulaşmış, 14.07.2005 tarihinde basım kararı alınmıştır. Makale ile ilgili tartışmalar 30.09.2006 tarihine kadar dergiye gönderilmelidir.

## Giriş

Son yıllarda kombinatoriyel problemlerin çözülmesinde sezgisel yöntemlerin kullanımları önemli oranda artmaktadır. Jones ve diğerlerine, (2002) göre sezgisel yöntemlerin popülaritesinin 1991 yılından itibaren hızlı bir şekilde artış göstermesinin nedenleri arasında da birincisi, hesaplama gücünün iyi olması ikincisi, dönüştürülebilir yönünün olmasıdır. Sezgisel yöntemlerin en büyük avantajları arasında çözüm zamanının sayım (enumeration) tekniğine göre çok kısa olması ve her tür problem için kolay bir şekilde entegre edilebilmesidir. Dezavantajları ise bu yöntemlerin optimum çözümü garanti etmemesi ve iyi çözüm verebilmesi için bir çok parametrenin uygun bir şekilde ayarlanması gerekliliğidir.

Bu sezgisel yöntemlerden biri olan Parçacık Sürü Optimizasyonu (PSO), kuş ve balık sürülerinin iki boyutlu hareketlerinden esinlenerek ilk olarak 1995 ve 1996 yıllarında Kennedy ve Eberhart tarafından geliştirilmiştir. PSO bireyler arasındaki sosyal bilgi paylaşımını esas alır. PSO'da arama işlemi genetik algoritmada olduğu gibi popülasyondaki bireyler tarafından ve belirlenen nesil sayısınca yapılır. Her bireye parçacık denir ve parçacıklardan oluşan popülasyona da sürü (swarm) ismi verilir. Her bir parçacık kendi pozisyonunu, bir önceki tecrübesinden yararlanarak bir önceki sürüdeki en iyi pozisyona doğru ayarlar (Eberhard ve Kennedy, 1995). Literatürde PSO bir çok alanda başarı ile uygulanmıştır. Bunlar şu şekilde sıralanabilir: Sipariş miktarı belirleme (lot sizing) problemi (Tasgetiren ve Liang, 2003), sinir ağları (neural network) eğitimi (Van den Bergh ve Engelbecht, 2000), akış tipi çizelgeleme problemleri (Tasgetiren vd., 2004a; 2004b), güç ve voltaj kontrolü (Yoshida vd., 2000; Abido, 2002), tek makine toplam pozitif gecikme problemi (Tasgetiren vd., 2004c), tedarik seçimi ve sıralama problemleri (Yeh, 2003) ve iş atama problemi (Salman vd., 2003).

Genel Atölye tipi çizelgeleme problemi, sonlu sayıda  $m$  tane tezgahta işlenmek üzere yine sonlu sayıda  $n$  tane işi, önceden belirlenen bir sıra ve kapasite kısıtlarını yerine getirerek, amaç fonksiyonunu optimum kılacak şekilde her bir

işlemin başlama zamanını belirlemek olarak tanımlanabilir. Bu problemde her bir işin tezgahlara uğrayacağı sıra farklıdır ve iki tip kısıtlamadan söz edilebilir. Birincisi, bir işin bir operasyonu bitmeden diğer operasyonunun başlayamamasıdır. Aynı zamanda bir iş bir anda sadece bir tezgah tarafından işlenebilir. İkincisi ise bir tezgah aynı anda sadece bir işin bir operasyonunu gerçekleştirir.

Atölye tipi çizelgeleme probleminde  $n$  iş  $m$  tezgah olmak üzere mümkün çizelgelerin sayısı  $(n!)^m$ 'dir. Bu sayı tezgah ve iş sayısının artması durumunda çok büyüyecektir ve bunlar arasında en iyi çizelgenin tespiti için harcanacak zaman aşırı fazla olacaktır. Pinedo (1995)'in ifadesiyle, eğer bir çizelgeleme problemini en iyi çözecek etkin, yani polinomsal bir zaman algoritması yoksa bu problemler NP-zor olarak sınıflandırılır. Genel atölye tipi çizelgeleme problemi NP-zor (Non-Deterministic Polynomial hard) problem sınıfına girer (Rinnooy, 1976).

Literatürdeki atölye tipi çizelgeleme problemlerinin çözüm yöntemleri, optimum ve yaklaşık çözümler olmak üzere ikiye ayırabilir. Optimum çözümü veren algoritmalar küçük çaplı problemler için uygundur. Örneğin dal sınır algoritması ve tamsayı programlama bu algoritmalar içerisinde yer almaktadır. Büyük çaplı problemlerde ise optimum çözümü bulmak çok zaman aldığından optimum ya da optimuma yakın çözümleri elde edebilmek için sezgisel algoritmalar kullanılır. Bu sezgisellerin başında değişen darboğaz yöntemi gelmektedir (Adams vd., 1988). Diğer önemli sezgisel yöntemler şu şekilde sıralanabilir: Genetik Algoritmalar (Dorndorf ve Pesch, 1995; Kumar vd., 1996; Zhou vd., 2001; Wang ve Zeng, 2001; Steinhofel vd., 2002; Murovec ve Suhel, 2004), Benzetim Tavlaması (Matsuo vd., 1988; Laarhoven vd., 1992; Kolonko, 1999; Satake vd., 1999, Aydın ve Fogarty, 2004), Tabu Arama (Tailard, 1994; Dell'Amico ve Trubian, 1993; Nowicki ve Smutnicki, 1996; Pezzella ve Merelli, 2000) ve Karınca Kolonileri (Blum ve Sampels, 2004).

Bu çalışmada, çözümü zor olan genel atölye tipi çizelgeleme problemlerinin çözümünde yeni bir

sezgisel yöntem olan PSO'yu değişken komşuluk arama yöntemi ile birlikte çalıştırıp, tamamlanma zamanı başarımlar ölçütüne göre performansları incelenmiştir. Bu çalışmanın ikinci bölümünde PSO modeli açıklanmakta bir sonraki bölümde de deneysel çalışma ve sonuçlar yer almaktadır.

### Atölye tipi çizelgeleme problemi için yerel aramalı parçacık sürü optimizasyonu modeli

Önerilen PSO modelinde ilk olarak kullanılacak parametreler belirlenir. İkinci olarak ise popülasyon değerleri rasgele belirlendikten sonra bu değerlere karşılık gelen amaç fonksiyon değeri hesaplanır. Amaç fonksiyon değeri en iyi olan parçacık küresel en iyi olarak atanır ve bu parçacık değerleri küresel komşular olarak saklanır. Aynı şekilde başlangıçtaki her bir parçacığın amaç fonksiyon değerleri yerel en iyiler olarak saklanıp parçacık değerleri de yerel komşular olarak saklanır.

```

Parametreleri Belirle
Her bir Parçacık İçin{
  -Başlangıç Pozisyon Vektörünü Oluştur
  -Başlangıç Hız Vektörünü Oluştur
  -Pozisyon Vektöründen Operasyon Sırası Elde Et
  -Operasyon Sırasından, Operasyon Tabanlı Gösterim Sırasını Elde Et
  -Amaç Fonksiyon Değerlerini Bul
  -Yerel En İyiyi Bul
}
-Küresel En İyiyi Bul
Yap {
  Her bir Parçacık İçin {
    -Başlangıç Pozisyon Vektörünü Oluştur
    -Başlangıç Hız Vektörünü Oluştur
    -Pozisyon Vektöründen Operasyon Sırası Elde Et
  }
  -Operasyon Sırasından, Operasyon Tabanlı Gösterim Sırasını Elde Et
  -Amaç Fonksiyon Değerlerini Bul
  -Yerel En İyiyi Bul
}
-Küresel En İyiyi Bul
-Küresel En İyiye Yerel Arama Uygula
} Durdurma Kriteri
    
```

Şekil 1. Atölye tipi çizelgeleme problemi için önerilen PSO modelinin yapısı

Bir sonraki nesli oluştururken hız vektörü, küresel ve yerel komşular değerleri kullanılır. Bu işlem bir durdurma kriterine kadar devam eder. Önerilen Parçacık Sürü Optimizasyonu modelinin benzetim kodu, Şekil 1'de verilmiştir. Bu yöntemde kullanılan temel unsurlar şu şekilde sıralanabilir.

**Pozisyon vektörü:**  $X_i^k$  ile ifade edilir, problem boyutu kadar eleman içerir. Burada problem boyutu  $j = n \times m$ 'dir. Pozisyon vektörü, problemi ifade etmekte kullanılır. Parçacığın pozisyon vektörünün sürü içerisinde gösterimi  $X_i^k = \{x_{i1}^k, x_{i2}^k, \dots, x_{i(n \times m)}^k\}$  şeklindedir. Burada yer alan  $x_{i1}^k$ ,  $k$ . iterasyonda  $i$ . parçacığın pozisyon vektöründeki birinci elemanını gösterir. Öyle ki  $\rho$  parçacık sayısını ifade etmektedir ( $i=1,2,\dots,\rho$ ).

**Hız vektörü:** Parçacığın bir sonraki konumunu belirleyen parametrelerinden biri olan hız vektörü,  $V_i^k$  ile ifade edilir. Parçacığın hız vektörü sürü içerisinde,  $V_i^k = \{v_{i1}^k, v_{i2}^k, \dots, v_{i(n \times m)}^k\}$  şeklinde gösterilir. Burada yer alan  $v_{i1}^k$ ,  $k$ . iterasyonda  $i$ . parçacığın hız vektöründeki birinci elemanını gösterir.

**Operasyon sırası vektörü:** Parçacığın pozisyon vektöründen elde edilir ve  $T_i^k$  ile ifade edilir. Sürü içerisinde,  $T_i^k = \{t_{i1}^k, t_{i2}^k, \dots, t_{i(n \times m)}^k\}$  şeklinde gösterilir.

**Operasyon tabanlı gösterim sırası vektörü:** Parçacığın operasyon sırası vektöründen elde edilir ve  $\Pi_i^k$  ile ifade edilir ve sürü içerisinde,  $\Pi_i^k = \{\pi_{i1}^k, \pi_{i2}^k, \dots, \pi_{i(n \times m)}^k\}$  şeklinde gösterilir.

**Yerel en iyi değer:**  $i$ . parçacığın o ana kadar elde edilmiş en iyi amaç fonksiyon değeridir ( $f(P_i)$ ). Sürü içerisinde yerel en iyilerin sayısı parçacık sayısı kadardır.

**Yerel en iyi komşular:** Yerel en iyilere karşılık gelen vektördür. Bu vektör değerlerini pozisyon vektöründen alır.  $i$ . parçacığın yerel en iyi komşularını gösterir.

şuları  $P_i = \{p_{i1}, p_{i2}, \dots, p_{ij}\}$  şeklinde ifade edilir.

Buradaki  $p_{i1}$ ,  $i$ . yerel en iyi vektörünün 1. değerini göstermektedir.

Küresel en iyi değer: Elde edilen en iyi amaç fonksiyon değeridir  $f(G)$ .

Küresel en iyi komşular: Küresel en iyinin pozisyon değerleridir.  $G = \{g_1, g_2, \dots, g_{(n \times m)}\}$ .

Atalet ağırlığı: Bir önceki hız vektörünün mevcut hız vektörü üzerindeki etkisini kontrol eden parametreye atalet ağırlığı denir ve  $w^k$  şeklinde gösterilir. Atalet ağırlığı, arama işleminin küresel veya yerel yapılmasını sağlayan bir parametredir. Bu parametrenin yüksek bir değer (0.5-2.5) seçilmesi, küresel bir şekilde arama gerçekleştirilmesini, küçük bir değer (0.1-0.5) seçilmesi ise, yerel bir arama gerçekleştirilmesini sağlar (Kennedy vd., 2001).

Tablo 1. Örnek: 2 iş-2 tezgah problemi

İşler	Operasyonlar	
	1	2
	İşlem Zamanları	
1	3	2
2	2	3
	Tezgah Sırası	
1	1	2
2	1	2

### Modelin gösterimi

Bu çalışmada, atölye tipi çizelgeleme probleminin gösterim şekli olarak, hem diğer gösterim yöntemlerine göre kolay olması hem de daha iyi performans vermesi (Ponnambalam vd., 2001) açısından Operasyon Tabanlı Gösterim (Cheng vd., 1996) yöntemi seçilmiştir. Tablo 2'de 2 iş-2 tezgah problemi için uyarlanmış parçacığın elemanları verilmiştir.

Normal PSO yönteminde parçacık sadece pozisyon ve hız vektöründen oluşmaktadır. Burada pozisyon vektörü, aday çözüm ifade etmemektedir. Dolayısıyla iki ara vektör olan operasyon sırası vektörü ve operasyon tabanlı gösterim sırası vektörü çözümü ifade etmek için oluşturulur.

Kullanılan gösterim şekli operasyon tabanlı olduğu için operasyon sayısı kadar vektör değeri yer alacaktır. Bu da  $n$  iş sayısını  $m$  de tezgah sayısını göstermek üzere problemin boyutu  $n \times m$ 'dir (Bierwirth vd., 1996).

Tablo 2. 2 iş-2 tezgah problemi için parçacık elemanları

Pozisyon	1	2	3	4
$X_i^k$	1.8	-0.99	3.01	0.72
$V_i^k$	-3.5	2.5	0.89	3.85
$T_i^k$	2	4	1	3
$\Pi_i^k$	1	2	1	2

Operasyon tabanlı gösterim sırasını elde etmek için  $X_i^k$ 'de oluşturulan değerler buldukları pozisyonlara göre küçükten büyüğe doğru sıralanarak  $T_i^k$  değerleri oluşturulur. Şöyle ki, Tablo 2'de en küçük pozisyon değeri  $x_{i2}^k = -0.99$ 'dir ve 2. pozisyonunda bulunmaktadır. Dolayısıyla  $t_{i1}^k$  değeri 2 olarak alınır. Aynı şekilde  $-0.99$  ten sonra gelen en küçük değer  $x_{i4}^k = 0.72$ 'dir ve 4. pozisyonunda yer almaktadır. Dolayısıyla  $t_{i2}^k$  değeri de 4 olur bu işlem  $X_i^k$ 'de yer alan bütün değerler için devam eder. Uygulanan bu işlem (Bean, 1994) tarafından önerilen rasgele gösterim metodunun permütasyonlu akış tipi çizelgeleme problemindeki uyarlamasıdır. Elde edilen  $T_i^k$  operasyon tabanlı sırası, (1)'de verilen formül kullanarak operasyon tabanlı gösterim sırasına dönüştürülür. (1)'de verilen formülde bir işin operasyon sayısı ( $m$ ) kadar ortaya çıkması için  $t_{ij}^k$  değeri toplam iş sayısına ( $n$ ) bölünüp çıkan değer bir üst sayıya yuvarlanır. Örneğin  $t_{i4}^k = 3$  değerinin operasyon tabanlı gösterim değeri  $\lceil (3/2) \rceil = \lceil 1.5 \rceil = 2$ 'dir.

$$\pi_{ij}^k = \left\lceil \frac{t_{ij}^k}{n} \right\rceil \quad (1)$$

Tablo 1’de  $\Pi_i^k$  operasyon tabanlı gösterim sırası verilmiştir. Bu değerler  $O_{nt}$ ,  $n$ . işin  $t$ . operasyonunu göstermek üzere  $[O_{11}, O_{21}, O_{12}, O_{22}]$  şeklinde ifade edilebilir. Dolayısıyla Tablo 1’deki problem için  $\Pi_i^k$  değerleri yardımıyla oluşturulan aktif çizelge Şekil 2’de görülmektedir.

1. Tezgah	1.iş		2.iş					
2. Tezgah				1.iş	2.iş			
Zaman	1	2	3	4	5	6	7	8

Şekil 2.  $\Pi_i^k$  değerlerinden elde edilen aktif çizelge

### Modelin işleyişi

Geliştirilen PSO modelinin tüm hesaplama işlemleri şu şekilde özetlenebilir.

*Adım 1: Başlangıç değerlerini oluştur.*

- İterasyon sayısını 0’a eşitle ( $k=0$ ) ve parçacık sayısını belirle
- Başlangıç pozisyon vektörü değerlerini  $[-d_{max}, d_{max}]$  aralığında rastgele oluştur.  $X_i^0 = \{x_{i1}^0, x_{i2}^0, \dots, x_{i(m \times n)}^0\}$ . Bu değerler kesikli değil süreklidir. Diğer bir deyişle gerçek sayılardır.
- Başlangıç hız vektörü değerlerini  $[-V_{max}, V_{max}]$  aralığında rastgele oluştur.  $V_i^0 = \{v_{i1}^0, v_{i2}^0, \dots, v_{i(m \times n)}^0\}$ . Bu değerler de gerçek sayılardır.
- Pozisyon vektörünü kullanarak operasyon sırası vektörünü oluştur. Başka bir deyişle sürekli olan başlangıç pozisyon değerlerini kesikli hale dönüştür.  $T_i^0 = [t_{i1}^0, t_{i2}^0, \dots, t_{i(m \times n)}^0]$ .
- Elde edilen operasyon sırası vektörünü (1)’deki formül yardımıyla operasyon tabanlı gösterim sırasına dönüştür.  $\Pi_i^0 = [\pi_{i1}^0, \pi_{i2}^0, \dots, \pi_{i(m \times n)}^0]$
- Amaç fonksiyon değerini elde edilen  $\Pi_i^k$

vektörü yardımıyla hesapla.  $f_i^k = f(\Pi_i^k)$ .

- Her bir parçacığın yerel en iyi değerlerini hesapla. Başlangıç parçacıklarının en iyi değerine  $P_i$  denirse bu değer başlangıçta  $P_i = X_i^0$  değerine eşittir.
- $P_i = \{p_{i1} = x_{i1}^0, p_{i2} = x_{i2}^0, \dots, p_{ij} = x_{ij}^0\}$  ve bu değerlere karşılık gelen amaç fonksiyonu değerleri ise  $f_i^{pb}$ ’dir.
- En iyi amaç fonksiyon değerine sahip olan parçacığı küresel en iyi olarak al ve pozisyon değerlerini de küresel en iyi komşular olarak sakla.  $f_b = \min\{f_i^0\}$ ,  $b \in \{i; i=1,2,\dots,\rho\}$  ve,  $G = \{g_1 = x_{b1}, g_2 = x_{b2}, \dots, g_n = x_{b(n \times m)}\}$  amaç fonksiyon değeri de  $f^{gb} = f_b$ ’dir.

*Adım 2: İterasyon sayısını güncelleştir*

- $k=k+1$

*Adım 3: Atalet ağırlığını güncelleştir.*

- $w^{k+1} = w^k \times \beta$

*Adım 4: Hız vektörünü güncelleştir.*

- $v_{ij}^{k+1} = (w^{k+1} \times v_{ij}^k) + c_1 \times r_1 \times (p_{ij} - x_{ij}^k) + c_2 \times r_2 \times (g_j - x_{ij}^k)$  Burada belirtilen  $r_1$  ve  $r_2$  (0,1) arasında yer alan rassal değerler,  $c_1$  ve  $c_2$  ise sosyal (social) ve kavramsal (cognitive) parametre değerlerini ifade eder. Elde edilen  $v_{ij}^k$  değeri (2)’deki formül yardımıyla kısıtlandırılır.

$$h(v_{ij}^i) = \begin{cases} V_{max}, & \text{eğer } v_{ij}^i > V_{max} \\ v_{ij}^i, & \text{eğer } |v_{ij}^i| \leq V_{max} \\ -V_{max}, & \text{eğer } v_{ij}^i < -V_{max} \end{cases} \quad (2)$$

*Adım 5: Pozisyon vektörünü güncelleştir.*

- $x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1}$

*Adım 6: Operasyon sırasını oluştur.*

- Pozisyon vektörü değerlerini kullanarak operasyon sırası vektörünü oluştur.  
 $T_i^k = [t_{i1}^k, t_{i2}^k, \dots, t_{ij}^k]$

*Adım 7: Operasyon tabanlı sırayı oluştur.*

- $\Pi_i^k = [\pi_{i1}^k, \pi_{i2}^k, \dots, \pi_{ij}^k]$

*Adım 8: Değerlendir.*

- Güncelleştirilmiş sürü içerisinde yer alan her bir parçacığın  $\Pi_i^k$  değerlerini kullanarak yeni amaç fonksiyon değerlerini bul.  $f_i^k = f(\Pi_i^k)$

*Adım 9: Yerel en iyi değerlerini güncelleştir.*

- Güncelleştirilen her bir parçacığı bir önceki amaç fonksiyonu değeri ile karşılaştır. Bir öncekinden daha iyi ise parçacığın değerlerini güncelleştir, aksi takdirde aynısını al. Eğer  $f_i^k < f_i^{pb}$ , ise  $P_i = X_i^k$  ve  $f_i^{pb} = f_i^k$  'dir.

*Adım 10: Küresel en iyi değeri güncelleştir.*

- Güncelleştirilen her bir parçacığı bir önceki sürüdeki amaç fonksiyonu değeri en iyi olan ile karşılaştır. Bir öncekinden daha iyi ise parçacığın değerlerini güncelleştir, aksi takdirde aynısını al.  $f_b^k = \min\{f_i^{pb}\}$ , Eğer  $f_b^k < f^{gb}$  ise  $G = X_b^k$  ve  $f^{gb} = f_b^k$  'dir.

*Adım 11: Küresel en iyiyi yerel arama uygula.*

- Küresel en iyi parçacığının operasyon tabanlı gösterim sırasına değişken komşuluk arama yöntemi uygula

*Adım 12: Durdurma kriteri.*

- Belirtilen maksimum iterasyon sayısına veya işlem zamanına erişildiğinde dur aksi halde 2. adıma git.
- Modelin komşuluk yapısı

Komşuluk yapısı mevcut bir çözümü değiştirerek yeni çözümler üretme şeklidir. Atölye tipi çizelgeleme problemlerinde bu tür bir yapı mevcut bir çizelgeden, komşuluk yapısının içe-

riğine göre yeni çizelgeler üretmek için kullanılır ve bu tür çizelgelere komşu çizelgeler denmektedir. Dolayısıyla yerel algoritma uygulamadan önce komşuluk yapısının belirlenmesi gerekmektedir. Modelde komşuluk yapısı pozisyon vektörüne veya operasyon tabanlı sıraya uygulanabilir. Yerel Arama yöntemi olarak ise basit ekle ve değiştir yöntemini içeren değişken komşuluk arama (VNS) yöntemi seçilmiştir.

Önerilen modelde komşuluk yapısı şu şekilde açıklanabilir. Örneğin Tablo 3'te pozisyon vektöründeki ikinci pozisyonda bulunan  $x_{i2}^k = -0.99$  değeri beşinci pozisyona eklendiğinde yeni bir çizelge elde edilmiş olur (Ekle Yöntemi). Görüldüğü gibi amaç fonksiyon değeri  $\Pi_i^k$  vektörü yardımıyla bulunmaktadır. Yerel arama yöntemi  $X_i^k$  vektörüne uyguladığında amaç fonksiyonundaki değişimi görmek için  $T_i^k$  ve  $\Pi_i^k$  değerlerinin bulunması gerekmektedir. Dolayısıyla bu işlem zaman harcayacağı için algoritmanın performansı düşecektir. Ancak yerel arama işlemi  $\Pi_i^k$  değerleri üzerinde yapılırsa daha az zamanda gerçekleştirilmiş olacaktır. Fakat bu işlem yapıldığında da en son elde edilen  $\Pi_i^k$  değerlerine karşılık gelen  $X_i^k$  ve  $T_i^k$  değerleri değişmiş olacaktır. Bu karışıklığı da engellemek için yerel arama ile elde edilen en son  $\Pi_i^k$ 'ye karşılık gelen  $X_i^k$  ve  $T_i^k$  vektörleri bulunur.

Bu işlem şu şekilde açıklanabilir. Tablo 4'den de görüldüğü gibi operasyon tabanlı gösterim sırası vektöründeki 1 ve 2 yer değiştirdiği halde (Değiştirme Yöntemi) bunlara karşılık gelen 2, 3 ve -0.99, 3.01 değerlerinin yeri değişmemiştir. Bir sonraki iterasyonda küresel en iyi parçacığının pozisyon vektörü değerleri kullanılacağı için elde edilen bu son operasyon sırası vektörü değerleri pozisyon değerlerine karşılık gelmemektedir. Dolayısıyla yerel arama işlemi bittiğinde operasyon tabanlı sıraya karşılık gelen  $T_i^k$  ve  $X_i^k$  değerleri yerine yazılır (Düzenle Fonksiyonu ile). Tablo 5'te bu işlemler görülmektedir.

Tablo 3. Pozisyon vektöründe komşuluk yapısı (Ekle)

Pozisyon	1	2	3	4
$X_i^k$	1.8	<b>-0.99</b>	3.01	0.72
$V_i^k$	-3.5	2.5	0.89	3.85
$T_i^k$	2	4	1	3
$\Pi_i^k$	1	2	1	2
$X_i^k$	1.8	3.01	0.72	<b>-0.99</b>
$V_i^k$	-3.5	2.5	0.89	3.85
$T_i^k$	4	3	1	2
$\Pi_i^k$	2	2	1	1

Tablo 4. Yerel aramanın operasyon tabanlı gösterim sırasına uygulanışı (önce)

Pozisyon	1	2	3	4
$X_i^k$	1.8	<b>-0.99</b>	<b>3.01</b>	0.72
$V_i^k$	-3.5	2.5	0.89	3.85
$T_i^k$	<b>2</b>	4	1	<b>3</b>
$\Pi_i^k$	<b>1</b>	2	1	<b>2</b>
$X_i^k$	1.8	<b>-0.99</b>	<b>3.01</b>	0.72
$V_i^k$	-3.5	2.5	0.89	3.85
$T_i^k$	<b>2</b>	4	1	<b>3</b>
$\Pi_i^k$	<b>2</b>	2	1	<b>1</b>

Tablo 5. Yerel aramanın operasyon tabanlı gösterim sırasına uygulanışı (sonra)

Pozisyon	1	2	3	4
$X_i^k$	1.8	<b>-0.99</b>	<b>3.01</b>	0.72
$V_i^k$	-3.5	2.5	0.89	3.85
$T_i^k$	<b>2</b>	4	1	<b>3</b>
$\Pi_i^k$	<b>1</b>	2	1	<b>2</b>
$X_i^k$	1.8	<b>3.01</b>	<b>-0.99</b>	0.72
$V_i^k$	-3.5	2.5	0.89	3.85
$T_i^k$	<b>3</b>	4	1	<b>2</b>
$\Pi_i^k$	<b>2</b>	2	1	<b>1</b>

Diğer bir deyişle  $\Pi_i^k$  değerlerine yerel arama uygulama sonucu  $\pi_{i1}^k = 1$  ve  $\pi_{i4}^k = 2$  yer değişmiştir. Bunlara karşılık gelen operasyon sırası ve pozisyon vektörü değerleri de değiştirilir. Yani operasyon sırası vektöründeki  $\tau_{i1}^k = 2 \rightarrow 3$  ve  $\tau_{i4}^k = 3 \rightarrow 2$  olarak, pozisyon vektöründeki  $x_{i2}^k = -0.99 \rightarrow 3.01$  ve  $x_{i3}^k = 3.01 \rightarrow -0.99$  olarak değiştirilir.

#### Yerel arama metodu

Önerilen modelde yerel arama işlemi, her bir iterasyon sonucunda küresel en iyi parçacığının operasyon tabanlı sıra vektörüne uygulanır. Yerel aramanın performansı seçilen iki tür komşuluk yapısına bağlıdır. Bunlar

- Operasyon tabanlı gösterim sırasındaki rassal olarak oluşturulan  $\eta$ . ve  $\kappa$ . değerlerinin yer değiştirilmesi ile yapılır ki,  $\eta \neq \kappa$  (Değiştirme).
- Operasyon Tabanlı sıradaki rassal olarak oluşturulan  $\eta$ . değer  $\kappa$ . değerinin arasına eklenir ki yine  $\eta \neq \kappa$ 'dir (Ekle).

Modelde yerel arama yöntemi olarak değişken komşuluk arama (variable neighborhood search) yöntemi kullanılmıştır (Mladenovic ve Hansen, 1997). Değişken komşuluk arama yönteminin yukarıda belirtilen komşuluk yapısı incelendiğinde *ekle+değiştir* ve *değiştir+ekle* olmak üzere iki gruba ayrılır. Bu modelde Ekle+Değiştir yöntemi incelenmiştir. Uygulanan bu yerel arama yönteminin benzetim kodu Şekil 3'te verilmiştir.

Burada yer alan  $S_0$  değeri küresel en iyi parçacığının operasyon tabanlı gösterim sırasını ifade etmektedir. Elde edilen en iyi değeri basit yerel arama yöntemleriyle bozup, değişken komşuluk arama metodu daha sonra uygulandığında daha iyi sonuçlar vermektedir (Besten vd., 2001; Stützle, 1998). Dolayısıyla o ana kadar elde edilen en iyi operasyon tabanlı sıra, rassal olarak  $n \times m$  arasında oluşturulan iki sayı ile ekleme ve değiştirme işlemi gerçekleştirilerek bozulur ( $Boz(S_0)$ ).

```

S0=J* Küresel En İyi Sırası (G)
S1 =Boz(S0)
Yap{
    döngü=0;
    maks_metot=2;
    sayaç=0;
    Yap{
        Eğer (sayaç==0) S2 =Değiştir(S1);
        Eğer (sayaç==1) S2 =Ekle (S1);
        Eğer f(S2) <= f(S1) {
            S1= S2;
            sayaç=0;
        }
        Değilse
            sayaç++;
    }sayaç < maks_metot;
}döngü<n×m

Eğer f(S1) <= f(S0) {
    G= S1;
    Düzenle(G);
}
    
```

Şekil 3. Uygulanan değişken komşu arama yönteminin benzetim kodu

Daha sonra *sayaç* şeklinde parametre tanımlanır. Bu parametre yardımıyla ilk önce değiştirme (interchange) işlemi yapılır. Amaç fonksiyonunda iyileşme var ise *sayaç* parametresi sıfıra eşitlenir. İyileşme yok ise bu parametre bir artırılır. *Sayaç=1* olduğu zaman bu defa ekle (insert) işlemi yapılır. Aynı şekilde iyileşme olduğu zaman bu işleme devam edilir, iki defa üst üste iyileşme olmadığı zaman ise tekrar değiştirme işlemi gerçekleştirilir. Bu işlem  $n \times m$  defa tekrar edilir. Sonuçta elde edilen operasyon tabanlı gösterim sırasına karşılık gelen operasyon sırası ve pozisyon vektörü değerleri *Düzenle(G)* fonksiyonu yardımıyla güncelleştirilir.

### Deneysel çalışma ve sonuçları

Yukarıda aşamaları belirtilen ve C programında yazılmış yerel aramalı PSO modeli; Tablo 6'da verilen PSO parametreleri kullanılarak Intel P4 2.6 Ghz, 512 RAM kapasiteli bilgisayarda, literatürdeki bazı zor test problemlerine 20 defa tekrar edecek şekilde uygulandığında Tablo 7'de verilen sonuçlar elde edilmiştir.

Bu modelde, optimumdan sapmaları göstermek için ortalama bağıl hata (Mean Relative Percent Deviation) ele alınmıştır. Ortalama bağıl hatanın bulunabilmesi için öncelikle bağıl hatanın bulunması gerekmektedir. Bağıl hatanın formülü (3)'te verilmiştir.

$$\text{Bağıl Hata} = \text{RE} = \frac{S - O}{O} \times 100 \quad (3)$$

Burada yer alan S değeri önerilen algoritmanın bulunduğu en iyi amaç fonksiyon değerini, O ise problemin optimum çözüm değerini veya şu ana kadar bilinen en iyi değerini göstermektedir. Ortalama bağıl hata (MRE) ise toplam bağıl hatalarının problem sayına bölümüyle elde edilir.

Tablo 6. Önerilen PSO modelinde kullanılan parametreler

Parametreler	
Popülasyon Sayısı ( $\rho$ )	$n \times m \times 2$
Başlangıç Atalet Ağırlığı ( $w$ )	0.9
Azaltma Fonksiyonu ( $\beta$ )	0.975
Pozisyon Vektör Aralığı ( $[d_{min}, d_{max}]$ )	[0,10]
Hız Vektör Aralığı ( $[V_{min}, V_{max}]$ )	[-4,+4]
Tekrar (Replication) Sayısı	20
Sosyal ve Kavramsal Değerler	2

Literatürdeki çalışmalarda genellikle algoritmanın performansını ölçmek için zor kabul edilen (Beasley, 2004) ve Tablo 7'de görülen zor test problemleri kullanılır. Bu test problemlerinde en iyi sonuç veren sezgisel yöntemler şu şekilde sıralanabilir;

- DT, Tabu Arama (Dell'Amico, 1993).
- TSAB, Tabu Arama (Nowicki ve Smunitcki, 1996).
- SB-RGLS2, Darboğaz yordamı tabanlı yerel arama yöntemi (Balas ve Vazacopoulos, 1998).

Tablo7 Yerel aramalı PSO algoritması ile elde edilen sonuçların diğer algoritmaların sonuçlarıyla karşılaştırılması

Problem	n	m	En İyi Değer	TSAB		SB-RGLS2		dESA		KSA		ACO GSS		GA TS		PSO	
				UB	RE	UB	RE	UB	RE	UB	RE	UB	RE	UB	RE	UB	RE
FT10	10	10	930	930	0.00	930	0.00	-	-	-	-	-	-	930	0.00	930	0.00
LA02	10	5	655	655	0.00	655	0.00	-	-	-	-	-	-	-	-	655	0.00
LA19	10	10	842	842	0.00	842	0.00	-	-	842	0.00	-	-	842	0.00	842	0.00
LA21	15	10	1046	1047	0.10	1046	0.00	1046	0.00	1047	0.10	1047	0.10	1046	0.00	1047	0.10
LA24	15	10	935	939	0.43	935	0.00	935	0.00	938	0.32	944	0.96	935	0.00	935	0.00
LA25	15	10	977	977	0.00	977	0.00	977	0.00	977	0.00	977	0.00	977	0.00	977	0.00
LA27	20	10	1235	1236	0.08	1235	0.00	1240	0.40	1236	0.08	1243	0.65	1235	0.00	1235	0.00
LA29	20	10	1152	1160	0.69	1164	1.04	1176	1.99	1167	1.30	1168	1.39	1153	0.09	1164	1.04
LA36	15	15	1268	1268	0.00	1268	0.00	-	-	1268	0.00	-	-	1268	0.00	1268	0.00
LA37	15	15	1397	1407	0.72	1397	0.00	-	-	1401	0.29	-	-	1397	0.00	1397	0.00
LA38	15	15	1196	1196	0.00	1196	0.00	-	-	1201	0.42	1196	0.00	1196	0.00	1196	0.00
LA39	15	15	1233	1233	0.00	1233	0.00	-	-	-	-	-	-	1233	0.00	1233	0.00
LA40	15	15	1222	1229	0.57	1224	0.16	1228	0.49	1226	0.33	1228	0.49	1222	0.00	1224	0.16
<b>Ortalama</b>					<b>0.20</b>		<b>0.09</b>		<b>0.48</b>		<b>0.28</b>		<b>0.51</b>		<b>0.01</b>		<b>0.1</b>

- KSA, Benzetim Tavlama (Kolonko, 1999).
- TSSB, Tabu Arama (Pezzella ve Merelli, 2000).
- ACO\_GSS, Tabu arama tabanlı karınca kolonileri yöntemi (Blum ve Sampel, 2004).
- dESA, Evrimsel benzetim tavlama (Aydin ve Fogarty, 2004).
- GA\_TS, Genetik algoritma tabanlı tabu arama yöntemi (Murovec ve Suhel, 2004).

Tablo 7'den de görüldüğü gibi Ortalama bağıl hatalara bakıldığında PSO (0.10) yönteminin, TSAB (0.20), dESA (0.48), KSA (0.28) ve ACO\_GSS (0.51)'den daha iyi olduğu, SB-RGLS2 (0.09)'e göre ise eşdeğer seviyede olduğu görülmektedir. GA\_TS (0.01) ise PSO'ya göre bir üstünlük sağlamaktadır. Murovec ve Suhel makalesinde elde edilen bu değerlerin 5000 defa tekrar sonucu elde edildiğini ifade

etmektedir ki bu da önemli bir dezavantajı göstermektedir.

Bilgisayar işlem zamanları açısından incelendiğinde ise, diğer algoritmaların kullandıkları bilgisayarlar birbirinden farklı olduğundan işlem zamanları verilmemiştir.

### Sonuç ve öneriler

Bu çalışmada, zor problemler sınıfına giren ve üretimde önemli bir yeri teşkil eden atölye tipi çizelgeleme problemlerinin çözümünde alternatif yeni sezgisel yöntem olan Parçacık Sürü Optimizasyonu ile bir model geliştirilmiştir.

Önerilen PSO modelinin sonuçları, literatürde iyi sonuçlar veren diğer sezgisel yöntemlerin (benzetim tavlama, tabu arama, karınca kolonileri ve genetik algoritma) sonuçlarıyla ortalama bağıl hata dikkate alınarak karşılaştırılmış, PSO modelinin genel olarak diğer algoritmalarla eşit seviyede veya daha iyi olduğu görülmüştür.

PSO modelinin diğer sezgisel yöntemlere göre avantajları;

- Kullanılan komşuluk yapısının basit ekleme veya değiştirme işlemleri yapılarak gerçekleştirilmesiyle, karmaşık komşuluk yapılarını kullanmadan da daha iyi sonuçlar elde edilebileceğini göstermesi.
- Daha az parametre içermesi ve program yapısının daha kolay oluşturulup diğer çizelgeleme problemlerine kısa sürede entegre edilebilmesi.

Bu çalışmanın literatüre katkıları ise;

- Partikül Sürü Optimizasyonu atölye tipi çizelgeleme problemlerine literatürde ilk olarak uygulanmıştır.
- Atölye tipi çizelgeleme problemi için permütasyonlu sıradan operasyon tabanlı gösterim sırası elde edilerek literatüre yeni bir gösterim metodu kazandırılmıştır.

şeklinde özetlenebilir.

Gelecekteki çalışma konuları ise şu şekilde sıralanabilir: Birincisi benzetim tavlama ve tabu arama gibi meta sezgisel yöntemlerin geliştirilen gösterim metodu ile performansları incelenip PSO yöntemiyle karşılaştırma yapılabılır. İkinci olarak ise önerilen PSO yönteminde sürekli değişkenler yerine kesikli değişkenler kullanılıp yeni bir model geliştirilebilir.

### Kısaltmalar

UB :Belirtilen algoritma tarafında bulunan en iyi değer

TSAB :Nowicki ve Smutnicki'nin (Nowicki ve Smutnicki, 1996) tabu arama yöntemi

SB-RGLS2:Balas ve Vazacopoulos (Balas ve Vazacopoulos, 1998) tarafından geliştirilen darboğaz yordamı tabanlı yerel arama yöntemi

dESA :Aydin ve Fogarty (Aydin ve Fogarty, 2004) tarafından geliştirilen evrimsel benzetim tavlama

KSA :Kolonko'nun (Kolonko, 1999) benzetim tavlama yöntemi

ACO\_GSS :Blum ve Sampels tarafından geliştirilen (Blum ve Sampel, 2004) tabu arama tabanlı karınca kolonileri metodu

GA\_TS :Murovec ve Suhel tarafından (Murovec ve Suhel, 2004) geliştirilen genetik algoritma tabanlı tabu arama yöntemi

PSO :Yerel Aramalı PSO modeli

### Kaynaklar

Abido, M. A., (2002). Optimal power flow using particle swarm optimization, *Electrical Power and Energy Systems*, **24**, 563-571.

Adams, J., Balas., E.,Zawack, D., (1988). The shifting bottleneck procedure for the job shop scheduling. *Management Science*, **34**, 391-401.

Aydin, M. E., Fogarty, T. C., (2004). A distributed evolutionary simulated annealing algorithm for combinatorial optimisation problems, *Journal of Heuristics*, **10**, 269-29

Balas, E., Vazacopoulos, A., (1998). Guided local search with shifting bottleneck for job shop scheduling problem, *Management Science*, **44(2)**, 262-275.

Bean, J. C., (1994). Genetic algorithm and random keys for sequencing and optimization, *ORSA Journal on Computing*, **6**, 2, 154-160.

Beasley, J. E., (2004). OR Library.  
<http://www.brunel.ac.uk/depts/ma/research/jeb/info.html>

Besten, M., Stutzle, T., Dorigo, M., (2001). Design of iterated local search algorithms, *Proceedings, Evolutionary Workshops, LNCS, Berlin*.

Bierwirth, C., Mattfeld, D., Kopfer, H., (1996). A generalized permutation approach to job shop scheduling with genetic algorithms, *OR-Spektrum*, **17**, 87-92.

Blum, C., Sampels, M., (2004). An ant colony optimization algorithm for shop scheduling problems, *Journal of Mathematical Modeling and Algorithms*, **3**, 285-308.

Cheng, R., Gen M., Tsujimura, Y., (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms-I. Representation, *Computers and Industrial Engineering*, **30**, 4, 983-997.

Dell'Amico, M., Trubian,M., (1993). Applying tabu search to the job shop scheduling problem, *Annals of Operations Research*, **41**, 231-252.

Dondorf, U., Pesch, E., (1995). Evolution based learning in a job shop scheduling environment, *Computers and Operations Research*, **22**, 25-40.

Eberhard, R. C., Kennedy, J., (1995). A new optimizer using particle swarm theory. *Proceedings, 6<sup>th</sup> International Symposium on Micro Machine and Human Science, Nagoya, Japan*.

Jones, D. F., Mirrazavi, S.K., Tamiz M., (2002). Multi-objective meta-heuristics:An overview of

- the current state-of-the-art, *European Journal of Operational Research*, **137**, 1-9.
- Kennedy, J., Eberhard, R.C., Shi, Y., (2001). *Swarm Intelligence*. San Mateo, Morgan Kaufmann.
- Kolonko, M., (1999). Some new results on simulated annealing applied to the job shop scheduling problem. *European Journal of Operational Research*, **113**, 123-136.
- Kumar, N. S. H., Srinivasan G., (1996). A Genetic Algorithm for Job Shop Scheduling-A case Study. *Computers in Industry*, **31**, 155-160.
- Laarhoven, P. J., Aarts, M. V., Lenstra, J. K., (1992). Job shop scheduling by simulated annealing. *Operational Research*, **40**, 113-125.
- Matsuo, H., Suh, C. J., Sullivan, R. S., (1988). A controlled search simulated annealing method for the general job-shop scheduling problem. *PhD Thesis*, Graduate School of Business, University of Texas at Austin
- Mladenovic, N., Hansen, P., (1997). Variable Neighborhood Search. *Computers and Operations Research*, **24**, 1097-1100.
- Murovec, B., Suhel, P., (2004). A reparing technique for the local search of the job-shop problem. *European Journal of Operational Research*, **153**, 220-238.
- Nowicki, E., Smutnicki, C., (1996). A fast taboo search algorithm for the job shop problem, *Management Science*, **42**, 797-813.
- Pezzella, P., Merelli, E., (2000). A tabu search method guided by shifting bottleneck for the job shop scheduling problem. *European Journal of Operational Research*, **120**, 297-310.
- Pinedo, M., (1995). *Scheduling: Theory, algorithms and systems*, Prentice-Hall, New Jersey.
- Ponnambalam, S. G., Aravindan, P., Rao, P. S., (2001). Comparative evaluation of genetic algorithms for job-shop scheduling. *Production Planning and Control*, **12**, 560-574.
- Rinnooy, K., (1976). *Machine Scheduling Problems: Classification Complexity and Computations*. Martinus Nijhoff, The Hague.
- Salman, A., Ahmad, I., Al-Madani, S., (2003). Particle swarm optimization for task assignment problem, *Microprocessors and Microsystems*, **26**, 363-371.
- Satake, T., Morikawa, K., Takahashi, K., Nakamura, N., (1999). Simulated annealing approach for minimizing the makespan of the general job-shop. *International Journal of Production Economics*, **60**, 515-522.
- Steinhöfel, K., Albrecht, A., Wong, C.K., (2002). Fast parallel heuristic for the job shop scheduling problem, *Computers and Industrial Engineering*, **29**, 151-169.
- Stützle, T., (1998). Applying iterated local search to the permutation flowshop problem. *Technical Report*, Darmstadt, Germany, Darmstadt University of Technology, Computer Science Department.
- Tailard, E., (1994). Parallel taboo search techniques for the job shop scheduling. *ORSA Journal on Computing*, **16**, 2, 108-117.
- Tasgetiren, M. F., Liang, Y. C., (2003). A binary particle swarm optimization algorithm for lot sizing problem, *Journal of Economic and Social Research*, **5**, 2, 1-20.
- Tasgetiren, M. F., Liang, Y. C., Sevkli, M., Gencyilmaz, G., (2004a). Particle swarm optimization algorithm for makespan and maximum lateness minimization in permutation flowshop sequencing problem, *Proceedings*, 4<sup>th</sup> International Symposium on Intelligent Manufacturing Systems (IMS2004), Sakarya, Turkey, 431-441.
- Tasgetiren, M. F., Sevkli, M., Liang, Y. C., Gencyilmaz, G., (2004b). Particle swarm optimization algorithm for permutation flowshop sequencing problem, *Proceedings*, 4<sup>th</sup> International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS2004), LNCS 3172, Brussels, 382-390, Belgium.
- Tasgetiren, M. F., Sevkli, M., Liang, Y. C., Gencyilmaz, G., (2004c). Particle swarm optimization algorithm for single machine total weighted tardiness problem, *Proceedings*, 2004 Congress on Evolutionary Computation (CEC'04), 1412-1419, Portland, Oregon.
- Van den Bergh, F., Engelbecht, A. P., (2000). Cooperative learning in neural networks using particle swarm optimizers, *South African Computer Journal*, **26**, 84-90.
- Wang, L., Zheng, D., (2001). An effective hybrid optimization strategy for job-shop scheduling problems. *Computers and Industrial Engineering* **28**, 585-596.
- Yeh, L. W., (2003). Optimal procurement policies for multi-product multi-supplier with capacity constraint and price discount, *Master Thesis*, Department of Industrial Engineering and Management, Yuan Ze University, Taiwan.
- Yoshida, H., Kawata, K., Fukuyama, Y., Nakanishi, Y., (2000). A particle swarm optimization for reactive power and voltage control considering voltage security assessment, *IEEE Transactions on Power Systems*, **15**, 1232-1239.
- Zhou, H., Feng, Y., Han, L., (2001). The hybrid heuristic genetic algorithm for job shop scheduling, *Computers and Industrial Engineering*, **40**, 191-200.