

ESNEK İMALAT SİSTEMLERİNDE PARÇA SEÇİMİ VE MAKİNA YÜKLEME İÇİN UZUN DÖNEM HAFIZALI BİR TABU ARAMA ALGORİTMASI

Murat ARIKAN ve Serpil EROL

Endüstri Mühendisliği Bölümü, Mühendislik Fakültesi, Gazi Üniversitesi, Maltepe, 06570, Ankara.
marikan@gazi.edu.tr, serpiler@gazi.edu.tr

(Geliş/Received: 06.04.2009 ; Kabul/Accepted: 05.08.2009)

ÖZET

Bu çalışmada, EİS planlama problemleri arasında yer alan parça seçimi ve makina yükleme problemleri ele alınmış ve sıralı olarak çözülen iki amaçlı bir karışık tamsayı programlama modeli ile ifade edilmiştir. Ancak, problemin kombinatoriyal bir yapıya sahip olması, büyük boyutlu problemlerin matematiksel modeller yardımıyla çözümlerini güçleştirmektedir. Bu nedenle, problemin çözümü için yoğunlaştırma ve çeşitlendirme stratejileri destekli bir tabu arama algoritması geliştirilmiştir. En iyi parametre seti faktöryel deney tasarımı ile belirlenen algoritmanın etkinliği, rassal olarak üretilmiş farklı büyüklükteki problemler üzerinde test edilmiş ve elde edilen sonuçlar matematiksel model çözümleriyle karşılaştırılmıştır.

Anahtar Kelimeler: Esnek imalat sistemleri, parça seçimi, makina yükleme, tabu arama.

A LONG TERM MEMORY TABU SEARCH ALGORITHM FOR PART SELECTION AND MACHINE LOADING IN FLEXIBLE MANUFACTURING SYSTEMS

ABSTRACT

In this study, part selection and machine loading problems of FMS planning phase are handled and expressed as a bi-objective mixed integer programming model which is solved sequentially. Unfortunately, the combinatorial structure of the problem makes the solution difficult and time consuming for real-world size problems by using the mathematical models. Therefore, a tabu search algorithm supported by an intensification and a diversification strategy is developed to solve the problem. The performance of the algorithm, for which the best parameter set is determined by factorial design analysis, is tested on the random generated problems with different sizes. The results are compared with those of the mathematical model.

Keywords: Flexible manufacturing systems, part selection, machine loading, tabu search.

1. GİRİŞ (INTRODUCTION)

Artan uluslararası rekabet, bir çok işletmenin dikkatini verimlilik ve kalite gibi kritik kavramlara yönelmesine ve üretim prosesleri için yeni yaklaşımlar arayışına girmesine neden olmuştur. Düşük verimlilik ve kalite sorunları için sıkça tavsiye edilen çarelerden biri fabrikaların otomasyonudur. Esnek İmalat Sistemleri (EİS), orta hacim/orta çeşit üretimde görülen sorunları ortadan kaldırmak için geliştirilmiş otomasyona dayalı sistemlerden biridir. Ancak, EİS'lerin sağladığı üstünlüklerden yararlanabilmenin tek yolu, bu sistemlerin etkin ve verimli bir şekilde işletilmelerinden geçmektedir. Kusiak [1], bir EİS'yi ilgilendiren

problemleri tasarım ve işletim olmak üzere iki ana başlık altında toplamıştır. Tasarım problemleri, kurulmasına karar verilen bir EİS'yi ilgilendirirken işletim problemleri, üretime hazır haldeki bir sistem için alınacak kararları kapsamaktadır. İşletim safhasında ele alınması gereken ilk problem grubu planlamadır ve sistem üretime başlamadan önce çözülmesi gereken problemleri içerir.

Bu çalışmada, planlama aşamasının iki ana problemi olan parça tipi seçimi ve makina yükleme problemleriyle ilgilendirilmiştir. Parça tipi seçimi, belli bir performans ölçütüne göre, EİS'de bir sonraki üretim çevriminde aynı anda işlenebilecek parçaların belirlen-

mesi olarak tanımlanırken, makina yükleme de seçilen parçalara ait operasyonların ve gerektirdikleri kesici uçların makinalara atanması ile ilgilenir. Birbirine yakından ilişkili kısıtlara sahip bu iki problem literatürde de sıklıkla birlikte ele alınmışlardır. Problem, çoğunlukla karışık tamsayılı bir matematiksel model olarak formüle edilmesine rağmen, kombinatoriyal bir yapıya sahip olduğundan, probleme özel sezgiseller [2-6] ya da tavlama benzetimi [7-9], tabu arama [9-11], genetik algoritmalar [6, 12-15], yapay bağışıklık sistemleri [16], parçacık sürü optimizasyonu [17], karınca kolonileri [18] ya da tavlama benzetimi ve tabu arama melez algoritması [19] gibi meta sezgisel yöntemlerle çözülmeye çalışılmıştır. EİS planlama aşaması, aynı anda ya da sıralı olarak ele alınabilecek çok amaçlı bir yapıya sahiptir. Co ve arkadaşları [2], Srivastava ve Chen [7, 10], Mukhopadhyay ve ark. [8], Arıkan ve Erol [9], Kumar ve Shanker [12], Yang ve Wu [14], problemi tek amaçlı incelemişlerdir. Moreno ve Ding [3], Tiwari ve Vidyarthi [13], Sarma ve ark. [11], Swarnkar ve Tiwari [19], Tiwari ve ark. [6], Prakash ve ark. [16, 18] problemi iki amaçlı olarak düşünmüşler, ancak EİS planlama aşamasının önemli bir parçası olan takım magazini yüklerinin belirlenmesi problemini dikkate almamışlardır.

Bu çalışmada ise parça seçim, makina yükleme ve takım magazini yerleşimi problemleri birlikte ele alınmıştır. Literatürdeki bir çok çalışmadan farklı olarak planlama döneminin sabit değil değişken olduğu kabul edilmiştir. Değişken dönem metodunda, üretim periyodunun uzunluğu sınırlanmaz [20]. Burada üretim periyodu, seçilen parçaların üretim ihtiyaçlarının karşılanması ile kısıtlıdır. Ayrıca, EİS planlama problemlerinin hiyerarşik yapısı dikkate alındığında parça seçim problemi, makina yükleme probleminin önce çözümlenmelidir. Bu çalışmada sözkonusu problemler, Liang ve Dutta [21]'da olduğu gibi sıralı olarak çözülen, bir karışık tamsayılı programlama modeli olarak formüle edilmişlerdir. Ancak, problem kombinatoriyal bir yapıya sahip olduğundan ve detaylı planlama seviyesini ilgilendirdiğinden hızlı ve iyi sonuçlar bulan bir yöntemle ele alınmalıdır. Bu nedenle, problemin çözümü için uzun dönem hafızalı bir tabu arama algoritması geliştirilmiştir.

2. PROBLEMİN TANIMI VE FORMÜLASYONU (PROBLEM DESCRIPTION AND ITS FORMULATION)

İncelenen EİS, genel akışlı atelye tipi bir EİS'dir. Yani, bir iş sisteme herhangi bir makinadan girip yine herhangi bir makinadan çıkabilir. Sistemdeki makinaları birbirine bağlayan malzeme taşıma sistemi, bir iş parçasını atelyede herhangi bir makinadan diğer herhangi bir makinaya taşıyabilecek esneklikte tasarlanmıştır. Sistemde, sınırlı kapasiteye sahip takım magazini ve otomatik takım değiştiricilerle donatılmış m makina, üretilmeyi bekleyen, bir veya daha fazla operasyondan oluşan, i farklı parça vardır. Her parçaya ait bir parti büyüklüğü mevcuttur. Her operasyon ve operasyonun ait olduğu parçaya ait tüm talep en fazla

bir makinaya atanabilir. Her operasyon bir kesici uç kullanılarak gerçekleştirilebilir, bir operasyonun gerçekleştirilebilmesi için ilgili kesici ucun makinanın takım magazinine atanmış olması gerekir. Operasyonların gerçekleştirilebileceği makinalar, gerektirdikleri kesici uçlar, işlem zamanları biliniyor ve sabittir.

Çalışmada ele alınan problem için kullanılan yaklaşım parti (tek dönem/tek parti) yaklaşımıdır. Seçilen parçalara ait operasyonların ihtiyaç duyduğu kesici uçlar takım magazinlerine yerleştirildikten sonra, eldeki işler tamamlanıncaya kadar makinaların takım magazininin yükleri değiştirilemez. Çevrime atanan parçaların tamamlanmasıyla, sistemin yüklenmesi için yeni bir parça seti belirlenerek işlemlere devam edilir. Problemin notasyonları ve matematiksel formülasyonu aşağıda verilmiştir.

İndisler

- i : parça tipleri, $i=1,2,\dots,I$
 k : operasyonlar, $k=1,2,\dots,K$
 t : takım tipleri, $t=1,2,\dots,T$
 m : makinalar, $m=1,2,\dots,M$

Parametreler

- W_i : i parçasının ağırlığı
 P_{ikm} : i parçasının k operasyonunun m makinasındaki işlem zamanı
 D_i : i parçasının talebi
 T_i : i parçasının toplam işlem zamanı
 $(T_i = \sum_{k=1}^K \sum_{m=1}^M P_{ikm} \cdot D_i \cdot V_{ikm})$
 J_m : m makinasında işlenebilen operasyonlar kümesi
 $M(i,k)$: i parçasının k operasyonunun işlenebileceği makinalar kümesi
 C_m : m makinasının takım magazini kapasitesi
 S_t : t takımının makina takım magazininde kapladığı yer
 TM_{ikmt} : i parçasının k operasyonu m makinasında t takımıyla işlenebiliyorsa 1, aksi halde 0.

Karar değişkenleri

- V_{ikm} : i parçasının k operasyonu m makinasında işleniyorsa 1, aksi halde 0
 Y_i : i parçası üretilmek üzere seçilmişse 1, aksi halde 0
 R_{tm} : t takımı m makinasına atanmışsa 1, aksi halde 0
 O_m : m makinasına atanan işyükünün ortalama yükten yukarı sapma miktarı
 U_m : m makinasına atanan işyükünün ortalama yükten aşağı sapma miktarı

Formülasyon

Kurulan modelin matematiksel formülasyonu ve kısıtların açıklamaları aşağıda verilmiştir.

$$\text{Max} \sum_{i=1}^I W_i Y_i \quad (1a)$$

$$\text{Min} \sum_{m=1}^M (O_m + U_m) \quad (1b)$$

s.t.

$$M. \left(\sum_{(i,k) \in J_m} P_{ikm} \cdot D_i \cdot V_{ikm} + O_m - U_m \right) = \sum_{i=1}^I T_i \cdot D_i \cdot Y_i \quad \forall m \quad (2)$$

$$\sum_{m \in M(i,k)} V_{ikm} = Y_i \quad \forall i, \forall k \quad (3)$$

$$V_{ikm} \leq \sum_{t=1}^T TM_{ikmt} \cdot R_{tm} \quad \forall i, \forall k, \forall m \quad (4)$$

$$\sum_{t=1}^T s_t \cdot R_{tm} \leq C_m \quad \forall m \quad (5)$$

$$V_{ikm} \in (0,1) \quad \forall i, \forall k, \forall m \quad (6)$$

$$Y_i \in (0,1) \quad \forall i \quad (7)$$

$$R_{tm} \in (0,1) \quad \forall t, \forall m \quad (8)$$

$$U_m, O_m \geq 0 \quad \forall m \quad (9)$$

(1a) ile W_i 'ye yüklenen anlama göre, üretim oranı veya üretilecek parçalardan sağlanacak kar ya da fayda maksimize edilmektedir. Bunun yanında W_i , parçaların teslim tarihinin bir fonksiyonu da olabilir. Böylece teslim tarihi daha erken olan parçalara daha büyük ağırlıklar verilerek bir sonraki üretim çevrimine atanmaları kolaylaştırılabilir. (1b) ile ise üretilecek parçalar belli olduğunda, ilgili operasyonların mümkün olduğunca dengeli bir şekilde makinalara atanması sağlanır. Kısıt (2), işyükü dengesini tanımlamaktadır. Seçilen parçalara ait toplam işyükü

$T_i = \sum_{k=1}^K P_{ikm} \cdot D_i \cdot V_{ikm}$ ifadesine eşittir. Bu durumda sistemde mevcut her makinaya düşen ortalama işyükü $\frac{\sum_{i=1}^I T_i \cdot D_i \cdot Y_i}{M}$ formülü ile elde edilir. O_m ve U_m

değişkenleri ile, ilgili makinaya yüklenen miktarın ortalama işyükünden yukarı ve aşağı sapma miktarları bulunur. Kısıt (3), i parçası işlenmek üzere seçilmişse,

i parçasına ait her operasyonun bir makinaya atanmasını sağlar. Kısıt (4), i parçasının k operasyonu m makinasına atanmışsa gerektirdiği kesici ucun da m makinasına atanmasını sağlar. Kısıt (5), m makinasına atanan takımların kapladığı toplam yerin, o makinanın takım magazini kapasitesini geçmemesini sağlar. (6)-(9) no'lu kısıtlar ise pozitiflik şartlarını ve değişken tiplerini göstermektedir. Problem, ilk önce (1a) amaç fonksiyonu altında çözülür ve aynı anda işlenebilecek maksimum parça sayısı, bu parçalara ait operasyonların atandığı makinalar ve takım magazini yerleşimleri belirlenir. Daha sonra, ilk aşamada elde edilen parça seçim amacı modelin altına kısıt olarak eklenir ve model (1b) amacı altında tekrar çözdürülür. Bu aşamada operasyon atamaları ve takım magazini yükleri yeni amacı en iyileyecek şekilde değişebilir. Böylece, parça seçim amacı optimize edilirken en dengeli operasyon ataması da yapılmış olur.

3. ÖNERİLEN TABU ARAMA ALGORİTMASI (PROPOSED TABU SEARCH ALGORITHM)

Tabu Arama (TA), Glover [22, 23] tarafından kombinatoriyal problemlerin çözümü için önerilmiş yüksek seviyeli bir sezgisel programlama tekniğidir. Tabu Arama tanımlanan probleme uygulanırken alınan kararlar aşağıda açıklanmıştır.

Çözüm uzayı: Parça seçim ve makina yükleme problemi için en kritik kararlardan biri hangi takımların hangi makinalara atanacağını belirlemesidir. Makinaların takım magazini yükleri, makinalar tarafından gerçekleştirilebilecek işlemleri, dolayısıyla da işlemlerin atanabileceği makinaları sınırlar. Bu çalışmada önceliğe sahip olan problemimiz parça seçimi olduğundan ve her parçanın sadece bir takım seti ile işlenebildiğini varsaydığımızdan, makinaların mümkün olduğunca farklı takımlarla donatılması, birinci önceliğe sahip amaca ulaşmamıza yardımcı olacaktır. Bu nedenle problemin çözümü, ilgili takımın hangi makinaya atandığını gösteren t boyutlu bir diziyile tanımlanmıştır. Bu çözüm, her takımın sadece bir makinaya atanmasını gerektirmektedir. Böylece, her işlemin atanabileceği bir makina olacaktır ve takım magazini yükleri ile operasyonların atanabileceği makinalar belirlenebilecektir.

Amaç fonksiyonu: Modelin sıralı olarak eniyilenmesi gereken iki amacı vardır. Problemin çok amaçlı yapısını temsil etmek için amaçlar ilk önce oransal hale getirilmiş, sonra da bütünleşik bir fonksiyon olarak ifade edilmiştir. Ayrıca, uygun olmayan çözümleri de arama prosedürüne dahil etmek için kısıt

$$(5)'teki \text{uygunsuzluk miktarını} \left(\sum_{m=1}^M \left(\sum_{t=1}^T s_t \cdot R_{tm} - C_m \right) \right)$$

dikkate alan bir ceza fonksiyonu tanımlanmıştır. Kullanılan bütünleşik amaç fonksiyonu aşağıda verilmiştir:

$$MaxZ = \gamma \cdot \frac{\left(\sum_{i=1}^I W_i \cdot Y_i \right) - \alpha \cdot \left(\begin{array}{l} \text{toplam} \\ \text{uygunsuzluk} \end{array} \right)}{\sum_{i=1}^I W_i} - \frac{\sum_{m=1}^M (U_m + O_m)}{\sum_{i=1}^I T_i \cdot D_i \cdot Y_i} \quad (10)$$

Bütünleşik amaç fonksiyonunun ilk bölümü parça seçim amacına, ikinci bölümü ise makina yükleme amacına karşılık gelmektedir. Amaçlardan biri maksimizasyon (parça seçimi) diğeri minimizasyon (makina yükleme) olduğundan iki amaç değerinin arasındaki farkı enbüyükleme istenilen sonucu verecektir. Bütünleşik amacın, orjinal matematiksel modelin sıralı yapısını tam olarak temsil etmesini sağlamak için de parça seçim amacına bir ağırlık (γ) verilmiştir. Yapılan ön denemeler sonucu, $\gamma=2.5$ için modelin sıralı yapısının yeterli derecede ifade edilebildiği gözlenmiştir. Ayrıca, toplam uygunsuzluk değeri de bir ceza katsayısı (α) ile çarpılmıştır. Ceza katsayılarının belirlenmesi için Srivastava ve Chen [7]'in kullandığına benzer bir deneysel yaklaşımdan yararlanılmıştır. Bu yaklaşıma göre, ele alınan her bir problem için başlangıçta α 'nın alt sınırı (α_l) 0, üst sınırı (α_u) ise algoritmanın mutlaka uygun bir çözüme yakınsamasını sağlayacak kadar büyük bir değer olarak alınır. Daha sonra algoritma, $\alpha = (\alpha_l + \alpha_u)/2$ değeri ile farklı rassal sayılar kullanılarak 10 defa çalıştırılır. Eğer bu koşullardan biri ya da daha fazlası uygun olmayan bir çözüm verirse, $\alpha_l = \alpha$ olarak alınır ve α_u sabit tutulur. Aksi takdirde, $\alpha_u = \alpha$ dönüştürmesi yapılır ve α_l sabit tutulur. Bu işlemler, uygun çözümler veren minimum değerli ceza katsayısı bulunana kadar güncellenmiş α değerleri ile tekrarlanır.

Hareket mekanizması: Mevcut çözümden komşu bir çözüme geçişte artma, azalma ve değişim hareketlerinden yararlanılmıştır. Mevcut çözüme uygulanacak hareket rassal olarak, sırasıyla, 0.2, 0.2 ve 0.6 olasılıklarıyla seçilmektedir.

Tabu listeleri ve tabu süreleri: Yapılan hareketin tabu olarak tanımlanması ve yapılacak bir hareketin tabu olup olmadığının kontrolünde biri artma/azalma ($t \times m$ boyutlu) diğeri değişim ($t \times t$ boyutlu) için olmak üzere iki ayrı tabu listesi kullanılmıştır. Bu listeler, ilgili hareketlerin başlangıç iterasyonlarını kaydederek ve tabu süreleri ile birlikte aday hareketin tabu olup olmadığının belirlenmesini sağlarlar. Artma/azalma ve değişim hareketlerinin tabu süreleri (sırasıyla, $t_{add/drop}$ ve t_{swap}) birbirine eşit olarak alınmıştır.

Tabu yıkma kriteri: Bu çalışmada, amaca göre tabu yıkma kullanılmıştır. Buna göre tabu olan bir hareket,

mevcut iterasyona kadar bulunmuş en iyi çözümden daha iyi bir çözümün elde edilmesini sağlıyorsa, tabu olmasına rağmen gerçekleştirilmektedir.

Aday liste stratejisi: Mevcut çözümün tüm komşularını incelenmesi yöntemi kullanılmıştır. Başka bir deyişle, sırayla, çözümdeki tüm değişkenler için artma, azalma ya da değişim hareketlerinden biri gerçekleştirilir.

Çeşitlendirme stratejisi: Bu çalışmada aramanın çeşitlendirilmesini yani çözüm uzayındaki farklı bölgelerin araştırılmasını sağlamak amacıyla hareket değerlerinin değiştirilmesi stratejisi kullanılmıştır. Hareket değerlerinin değiştirilmesinde, uzun dönem hafızada tutulan "geçicilik sıklığı" bilgisinden yararlanılmıştır. Geçicilik sıklığı ile mevcut çözümden yeni bir çözüme geçiş sırasında, hangi takımın hangi makinaya atandığını gösteren değişkenin ($tool_mach[t]$) kaç kere değiştirildiğinin kaydı tutulmaktadır. Bu sayede, arama boyunca daha sık yapılan hareketler belirlenir ve seyrek gerçekleştirilen hareketler teşvik edilerek aramanın yönü değiştirilebilir. Geçicilik sıklığı gerçekleştirilen harekete göre aşağıdaki gibi güncellenmektedir:

- Artma/azalma hareketi için;
Örneğin; mevcut çözümde c takımı d makinasında e makinasına atanmış ise ($tool_mach[c] = d \rightarrow tool_mach[c]=e$) geçicilik sıklığı listesi $frequency[c, e]=frequency[c, e]+1$ olacak şekilde güncellenir.
- Değişim hareketi için;
Örneğin; mevcut çözümde c takımının atandığı e makinası ile d takımının atandığı f makinasını ifade eden değişkenler arasında bir değişim hareketi uygulanırsa ($tool_mach[c]=e, tool_mach[d]=f \rightarrow tool_mach[c]=f, tool_mach[d]=e$) geçicilik sıklığı listesi, $frequency[c, f]=frequency[c, f]+1$ ve $frequency[d, e]=frequency[d, e]+1$ olacak şekilde güncellenir.

Hareket değerlerinin değiştirilmesi, geçicilik sıklığının ceza bilgilerine dönüştürülmesi vasıtasıyla gerçekleştirilir. Her iterasyonda değiştirilmek üzere seçilen değişken ya da değişkenlere ait geçicilik sıklığı oranı bir ceza katsayısı ile çarpılarak orjinal amaç fonksiyonundan çıkarılır ve değiştirilmiş bir amaç fonksiyonu değeri elde edilir. Yapılan hareketin cinsine göre, eşitlik (11) ve (12)'de verilmiş olan değiştirilmiş amaç fonksiyonları, mevcut çözümün komşuları arasından en iyisinin belirlenmesinde kullanılır. Algoritma sırasında geçicilik sıklıkları ilk iterasyondan itibaren tutulurken, değiştirilmiş amaç fonksiyonuna göre seçim belli bir iterasyon limitinden ($iterfreqlim$) sonra yapılmaktadır. Bu çalışmada, sözkonusu limit 3000 olarak belirlenmiştir. Yukarıda verilen örneğe göre;

- Artma/azalma hareketlerinden sadece birinin gerçekleştirildiği durum için;

$$\text{Modified_f(x)} = f(x) - \text{penalty} * \left(\frac{\text{frequency(c,e)}}{\text{iterasyon}} \right) \quad (11)$$

- Değişim hareketinin gerçekleştirildiği durum için;

$$\text{Modified_f(x)} = f(x) - \text{penalty} * \left(\left(\frac{\text{frequency(c,f)}}{\text{iterasyon}} \right) + \left(\frac{\text{frequency(d,e)}}{\text{iterasyon}} \right) / 2 \right) \quad (12)$$

Bu eşitlikler ile çözüme daha çok girip çıkan değişkenlerin seçilmesi zorlaştırılırken, seyrek olarak yapılan hareketler de teşvik edilir.

Yoğunlaştırma stratejisi: Kullanılan yoğunlaştırma stratejisiyle, arama boyunca karşılaşılan ve iyi çözümler içerdiği gözlenen arama alanlarına dönüş yapılarak daha detaylı araştırılmaları amaçlanmaktadır. Buna göre ilk önce, algoritmanın, yukarıda anlatılan özelliklere sahip kısmı çalıştırılır ve arama boyunca bulunan çözümler sınırlı boyuttaki bir listeye kaydedilir. Bu liste elit çözümler listesi olarak adlandırılır. Çeşitlendirme stratejisini de içeren söz konusu kısım ile ilgili durdurma koşulu sağlandığında, listenin en başındaki çözüm (listedeki en iyi çözüm) başlangıç çözümü olarak alınarak algoritma tekrar çalıştırılır. Arama esnasında listedekilerden daha iyi bir çözüm bulunursa listeye eklenir. Liste uzunluğunu sabit tutmak için yeni çözümler eklendikçe daha düşük kaliteli çözümler listeden çıkarılır. Listeye kayıtlı her elit çözümlerle algoritma yeniden başlatılırken, kısa dönem hafıza bilgileri korunmaktadır. Bu işlemler elit listedeki tüm çözümler bitinceye ya da önceden belirlenmiş bir iterasyon sayısına ulaşıncaya kadar sürdürülür.

Durdurma koşulu: Algoritma iki durdurma koşuluna sahiptir. Algoritma, yeni bir elit çözümlerle başlatılmadan önce en iyi çözüm değişmeden belli bir iterasyon sayısının (*iterfark*) geçmesi gerekmektedir. Ancak, algoritmanın bütünü ya elit çözüm listesi boşaldığında ya da önceden belirlenmiş bir iterasyon sayısına (*maxiter*) ulaşıldığında durdurulmaktadır. Geliştirilen algoritma için *maxiter* 100000 olarak alınmıştır.

TurboPascal programlama dili ile kodlanan algoritmanın adımları Şekil 1’de verilmiştir.

4. DENEY TASARIMI ÇALIŞMASI (EXPERIMENTAL DESIGN)

Algoritmanın göstereceği performans, algoritma parametrelerine verilecek değerlerle yakından ilgilidir. Bu nedenle, algoritmanın en iyi sonuç verdiği parametre setinin belirlenmesi için bir deneysel çalışma yapılmıştır. Geliştirilen algoritmanın performansını ölçmek için makina sayıları 5-8, parça sayıları 20-40, operasyon sayıları 3-5, takım sayıları da 20-50 arasında olmak üzere 8 farklı problem büyüklüğü belirlenmiştir. Problem büyüklükleri sabit tutularak probleme ait parametreler, takımların uyumlu olduğu makinalar, operasyonların gerektirdikleri kesici uçlar tekdüze rassal dağılıma uygun olarak belirlenmiştir.

Geliştirilen algoritmada Eşitlik (10)’daki oransal bütünleşik amaç fonksiyonu kullanılmıştır. Ancak, matematiksel model sıralı olarak çözülmekte ve 2 ayrı amaç fonksiyonu elde edilmektedir. Algoritmanın etkinliğini matematiksel olarak belirleyebilmek için parça seçim ve makina yükleme modellerinin GAMS programının CPLEX çözücüsü ile çözümünden elde edilen sonuçlar Eşitlik (10) kullanılarak tek bir amaç fonksiyonuna dönüştürülmüştür. Bu dönüştürme işleminden sonra faktöryel tasarım için kullanılacak yanıt değişkeni tabu arama algoritmasından elde edilen çözüm değerlerinin GAMS bütünleşik amaç fonksiyonundan oransal sapması olarak belirlenmiştir.

Algoritmanın en iyi performans göstereceği parametre setini belirlemek için 3^k faktöryel tasarım kullanılmıştır. 8 test problemi içinden, küçük, orta ve büyük boyutlu olmak üzere 3 problem (problem 2, 5 ve 8) faktöryel tasarım için seçilmiş ve yanıt değişkeni bu 3 problemin sapmalarının ortalaması olarak alınmıştır. Faktöryel tasarımında gözönüne alınan faktörler ve düzeyleri Çizelge 1’de görülmektedir.

Tabu süreleri faktörünün ilk iki düzeyi statik, üçüncü düzeyi ise dinamikdir. Statik tabu süreleri artma/azalma ve değişim hareketleri için eşit alınmıştır. Dinamik tabu süreleri için de artma/azalma ve değişim tabu süreleri eşit kabul edilmektedir, ancak dinamikliği sağlamak amacıyla belli bir iterasyon sayısı geçtikten sonra tabu süreleri rassal olarak tekrar belirlenmektedir. Tabu süreleri yeniden hesaplanmadan geçecek iterasyon sayısı 200 alınmıştır. Çizelgede tabu sürelerinin, sırasıyla, minimum ve maksimum değerleri verilmiştir.

Faktör düzeylerinin oluşturduğu her kombinasyonda

Çizelge 1. Deney tasarımına dahil edilen faktörler ve düzeyleri (The factors included into the experimental design and their levels)

Faktör ismi	Düşük Düzey (-1)	Orta Düzey (0)	Yüksek Düzey (+1)
Tabu süreleri (tt)	3, 3	7, 7	3, 7
En iyi çözüm değişmeden geçecek iterasyon sayısı (iterfark)	10	100	1000
Elit liste uzunluğu (elitcozsay)	10	20	200
Çeşitlendirme için kullanılan ceza katsayısı (penalty)	1	5	10

Adım 1: Başlangıç değerlerinin verilmesi ($tt_{add/drop}$, tt_{swap} , $maxiter$, α , $iterfark$, $Elitcozsay$, $penalty$)

Adım 2: Başlangıç çözümü x' 'i üret, elit çözümler listesine kaydet.

Adım 3: Elit çözümler listesinin ilk sırasındaki çözümü mevcut çözüm olarak kaydet
($x=elitçözümli[1]$, $elitçözümli[1]=\phi$)

Adım 4: (mevcut çözümün komşuluğunun araştırılması)
 $komsuluk_best_f=0$; $modified_komsuluk_best_f=0$; $nn=1$

4.1: Mevcut x çözümünün nn . elemanından rassal olarak belirlenmiş h_k hareketi ile ulaşılabilen x' çözümünü üret, $nn=nn+1$.

4.2: Eğer $iter < iterfreqlim$ ise bütünlük oransal amaç fonksiyonunu hesapla ($butunlesik_f[X(h_k)]$) aksi halde değiştirilmiş bütünlük oransal amaç fonksiyonunu hesapla ($mod_butunlesik_f[X(h_k)]$).

4.3: Eğer h_k tabu değil veya $butunlesik_f[X(h_k)] > best_f$ ise
Eğer $iter < iterfreqlim$ ise
Eğer $butunlesik_f[X(h_k)] > komsuluk_best_f$ ise
 $komsuluk_best_f = butunlesik_f[X(h_k)]$, $X_{komsuluk} = X(h_k)$
{komşuluktaki en iyi çözümü güncelle}
aksi halde adım 4.1'e git.
aksi halde
Eğer $mod_butunlesik_f[X(h_k)] > mod_komsuluk_best_f$ ise
 $Modified_komsuluk_best_f = mod_butunlesik_f[X(h_k)]$,
 $komsuluk_best_f = butunlesik_f[X(h_k)]$, $X_{komsuluk} = X(h_k)$
{komşuluktaki en iyi çözümü güncelle}
aksi halde adım 4.1'e git.

4.4: Eğer $nn < t$ ise (o iterasyonda çözümün tüm elemanları incelenmemiştir) adım 4.1'e dön.

Adım 5: Eğer $Komsuluk_best_f > best_f$ ise en iyi çözümü güncelle ve elit çözüm olarak kaydet. $X_{eniye} = X_{komsuluk}$, $best_f = komsuluk_best_f$; $iterbest = iter$.

Adım 6: Mevcut çözümü güncelle,
 $X = X_{komsuluk}$; $f(x) = komsuluk_best_f$.

Adım 7: Geçicilik sıklığı listesini güncelle
 $frequency[h_k] = frequency[h_k] + 1$

Adım 8: Tabu listelerini güncelle
Eğer h_k artırma/azaltma hareketi ise $tabustart[h_k] = iter$, değişim hareketi ise $tabustartswap[h_k] = iter$, $iter = iter + 1$

Adım 9: Eğer $iter - iterbest < iterfark$ ise Adım 4'e git,

Adım 10: Elit listeyi güncelle (en iyi çözümden başlayarak elitcozsay tanesini sırala)

Adım 11: Global en iyi çözümü güncelle
Eğer $best_f > best_of_f$ ise o ana kadar bulunan en iyi çözümü güncelle,
 $X_{globaleniye} = X_{eniye}$, $iterbest_{global} = iter$,

Adım 12: Global durdurma koşulunu kontrol et
Eğer ($elitçözümli[1] = \phi$) veya ($iter > iterlim$) ise DUR aksi halde Adım 3'e git.

Şekil 1. Geliştirilen uzun dönem hafızalı Tabu Arama algoritmasının adımları (The steps of the long term memory Tabu Search algorithm)

farklı rassal sayılar kullanılarak 5'er deneme yapılmış ve her problem için ($3^4 * 5 =$) 405 olmak üzere toplamda ($3 * 405 =$) 1215 deneme gerçekleştirilerek algoritmanın en iyi parametre setinin belirlenmesi amaçlanmıştır. Seçilen faktörlerin algoritma performansı üzerinde etkin olup olmadığını belirlemek için varyans analizi (ANOVA) kullanılmıştır. Deney sonuçlarının varyans analizi varsayımlarını sağlayıp sağlamadıklarını test etmek için yapılan regresyon analizi sonucunda hata terimlerinin normal dağılımdan çok uzaklaşmadıkları görüldü de, varyansın sabit olmadığı tespit edilmiştir. Bu sebeple veri transformasyonu uygulanmış ve veriler üzerinde Log dönüşümü gerçekleştirilerek varsayımların doğruluğu gösterilmiştir. Varyans analizinde ana etkiler ve faktörlerin birbiriyle olan ikili, üçlü ve dördümlü ortak etkileri incelenmiş ve bu faktörlerin dikkate alınan düzeyleri arasında anlamlı bir farklılık olup olmadığı Duncan çoklu aralık testiyle değerlendirilmiştir. Çizelge 2'de varyans analizinin, Çizelge 3'te ise Duncan çoklu aralık testinin sonuçları verilmiştir.

Çizelge 2'deki P değerleri incelendiğinde, $\alpha=0.05$ anlamlılık düzeyinde, sadece $iterfark$ ve $elitcozsay$ faktörlerinin ana etkilerin algoritmanın performansı üzerinde etkin olduğu görülmektedir. İkili ortak etkilerden $iterfark$ 'ın $elitcozsay$ ile olan ikili ortak etkisi, üçlü ortak etkilerden ise tt , $iterfark$, $penalty$ ve tt , $elitcozsay$, $penalty$ arasındaki ortak etkiler anlamlı bulunmuştur. Ayrıca, dördümlü ortak etkinin de anlamlı olduğu ortaya çıkmıştır.

Çizelge 3 incelendiğinde ise, yine $\alpha=0.05$ anlamlılık düzeyinde, tabu süresi (tt) ve çeşitlendirme stratejisi için kullanılan ceza katsayısı ($penalty$) parametrelerinin düşük, orta ve yüksek düzeyleri arasında anlamlı bir farklılık gözlenmemiştir. En iyi çözüm değişmeden geçecek iterasyon sayısı ($iterfark$) ve elit çözüm sayısı ($elitcozsay$) parametrelerinin ise yüksek-orta düzeyleri anlamlı bir farklılığa sahip değilken düşük düzeyin, yüksek ve orta düzeye kıyasla daha kötü sonuç verdiği anlaşılmıştır. Tabloda faktör düzeyleri,

Çizelge 2. Varyans analizi sonuçları (ANOVA results)

Değişim kaynağı	Kareler Toplamı	Serbestlik Derecesi	Ortalama Kare	F değeri	P değeri
<i>tt</i>	,127	2	6,368E-02	2,282	,104
<i>iterfark</i>	6,728	2	3,364	120,565	,000
<i>elitcozsay</i>	10,683	2	5,342	191,447	,000
<i>penalty</i>	6,271E-02	2	3,135E-02	1,124	,326
2 ortak					
<i>tt*iterfark</i>	9,662E-02	4	2,416E-02	,866	,485
<i>tt*elitcozsay</i>	8,986E-02	4	2,247E-02	,805	,523
<i>tt*penalty</i>	3,916E-02	4	9,789E-03	,351	,843
<i>iterfark*elitcozsay</i>	7,760	4	1,940	69,529	,000
<i>iterfark*penalty</i>	5,906E-02	4	1,477E-02	,529	,714
<i>elitcozsay*penalty</i>	6,281E-02	4	1,570E-02	,563	,690
3 ortak					
<i>tt*iterfark*elitcozsay</i>	,332	8	4,149E-02	1,487	,161
<i>tt*iterfark*penalty</i>	,522	8	6,519E-02	2,336	,019
<i>tt*elitcozsay*penalty</i>	,440	8	5,500E-02	1,971	,049
<i>iterfark*elitcozsay*penalty</i>	,405	8	5,063E-02	1,815	,074
4 ortak					
<i>tt*iterfark*elitcozsay*penalty</i>	,867	16	5,420E-02	1,943	,016
Hata	9,040	324	2,790E-02		
Toplam	1834,710	405			

Çizelge 3. Deney tasarımında kullanılan faktör düzeyleri için Duncan çoklu aralık testi sonuçları (Duncan multiple range test results for the factor levels used in experimental design)

Faktörler	Düzeyler	Grup Ortalaması *	Test Sonuçları	
			Kombinasyonlar	Anlamlı Farklılık
1. <i>tt</i>	(3,3)	-2,1221	(3,3)-(3,7)	Yok
	(3,7)	-2,1160	(3,3)-(7,7)	Yok
	(7,7)	-2,0818	(3,7)-(7,7)	Yok
2. <i>iterfark</i>	1000	-2,2045	100-100	Yok
	100	-2,1909	1000-10	Var
	10	-1,9246	100-10	Var
3. <i>elitcozsay</i>	200	-2,2379	200-20	Yok
	20	-2,2042	200-10	Var
	10	-1,8778	20-10	Var
4. <i>penalty</i>	5	-2,1241	5-1	Yok
	1	-2,1003	5-10	Yok
	10	-2,0956	1-10	Yok

algoritmanın gösterdiği performans açısından iyiden kötüye doğru sıralanmışlardır.

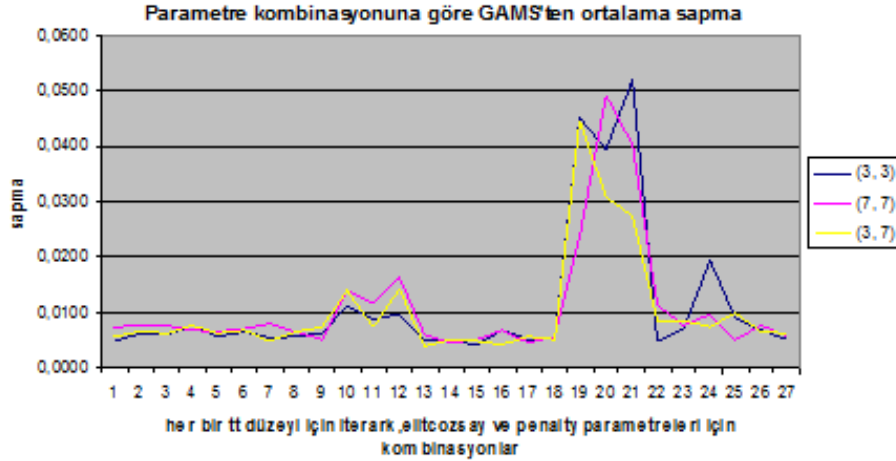
Varyans analizinden elde edilen sonuçlara göre algoritma parametreleri arasında karmaşık bir etkileşim vardır. İkili ortak etkilerden 1 tanesinin, üçlü ortak etkilerden 2 tanesinin ve dördü ortak etkinin anlamlı olduğu tespit edilmiştir. *tt* ve *penalty* faktörleri, tek başlarına performans üzerinde etkili görünmemelerine rağmen diğer iki ya da üç faktörle bir araya gelince etkili olabilmektedirler. Ayrıca, *iterfark-elitcozsay* ikili etkisinin anlamlı çıkması da incelemeye değerdir. Dolayısıyla, en iyi sonuç veren parametre setini belirlemek için, yukarıdaki testlere ek olarak, tüm parametre kombinasyonlarının ortalama sapma üzerindeki etkisi incelenmiştir. Şekil 2'de *iterfark*, *elitcozsay* ve *penalty* parametrelerinin tüm kombinasyonları için tabu süresi (*tt*) parametresinin düzeylerinde çizdirilen grafik görülmektedir

Şekil incelendiğinde en düşük ortalama sapma % 0,40 ile (3,7) tabu süresi düzeyindeki 13. parametre kombinasyonunda elde edilmektedir. Buna göre; algoritmanın en iyi performans gösterdiği parametre seti *tt* = (3,7), *iterfark* = 100, *elitcozsay* = 20, *penalty* = 1 olarak belirlenmiştir. Algoritmanın, belirlenen parametre seti ile, test problemleri üzerindeki performansı parça seçim ve makina yükleme problemleri için ayrı ayrı olmak üzere Çizelge 4 ve Çizelge 5'te görülmektedir.

Parça seçim amacı için algoritmanın performansı incelendiğinde, 6 no'lu test problemi hariç tüm problemlerde her denemede optimal parça seçim amacı elde edilmiştir. 6 no'lu problem için ise optimal çözümden sapma %0,083 oranındadır (Çizelge 4).

Makina yükleme amacı için, mutlak dengesizlik oranları yerine dengesizlik miktarının toplam işyüküne oranları alınarak karşılaştırma yapılmıştır. Çizelge 5'te algoritmanın en iyi parametre seti ile yapılan 5'er koşumun ortalama ve en iyi değerleri yanında, dengesizlik miktarlarının toplam işyükü miktarlarına

*Transforme edilmiş verilerin grup ortalamalarını göstermektedir.



Şekil 2. Parametre kombinasyonuna göre GAMS'ten ortalama sapmalar (Mean deviations from GAMS in respect of parameter combinations)

oranları da verilmiş ve GAMS çözümlerinden farkları hesaplanmıştır. Buna göre; GAMS ile optimal sonucu bulunan problemlerden 1 ve 3 için her denemede optimal sonuca ulaşılmıştır. 2 no'lu problemde 5 denemeden 3'ünde optimal sonuç elde edilmiştir. Diğer optimalleri bilinen 5 ve 7 no'lu problemlerde ise hiç bir denemede en iyi sonuç bulunamamıştır. Verilen süre kısıtı içerisinde optimalleri elde edilemeyen problemler dikkate alındığında ise, 6 no'lu problem için 5 denemeden 1'inde GAMS'ten daha iyi bir makina

yükleme amacına ulaşılmıştır. Tüm problemler dikkate alındığında algoritma ile elde edilen ortalama dengelesizlik oranlarının GAMS çözümlerinden farkı % 0.00 ile % 1,6 arasında değişmektedir. Ortalama fark hesaplandığında ise elde edilen sonuçların GAMS sonuçlarına göre % 0.79 daha kötü olduğu görülmektedir.

Çizelge 4. Algoritmanın en iyi parametre setiyle çalıştırılmasından elde edilen parça seçimi amacının GAMS çözümleriyle karşılaştırılması (Comparison of the part selection objective with GAMS results that are obtained from the algorithm by using the best parameter set)

Pr. No.	GAMS				TA _{İDV}			Ortalama Sapma GAMS- TA _{İDV} (%)
	m	i	k	t	Çözümü	Ort.	Eniyi	
1	5	20	3	20	369.0 ^a	369.0	369.0	0.000
2	5	20	5	30	287.0 ^a	287.0	287.0	0.000
3	5	30	3	30	411.0 ^a	411.0	411.0	0.000
4	5	30	5	40	454.0 ^a	454.0	454.0	0.000
5	8	30	3	30	768.0 ^a	768.0	768.0	0.000
6	8	30	5	40	480.0 ^a	479.6	480.0	0.083
7	8	40	3	40	691.0 ^a	691.0	691.0	0.000
8	8	40	5	50	883.0 ^a	883.0	883.0	0.000

^aOptimal çözüm elde edilmiştir

Çizelge 5. Algoritmanın en iyi parametre setiyle çalıştırılmasından elde edilen makina yükleme amacının GAMS çözümleriyle karşılaştırılması (Comparison of the machine loading objective with GAMS results that are obtained from the algorithm by using the best parameter set)

Pr. No.	GAMS				TA _{İDV}			GAMS'ten Dengesizlik oranı farkı (2)-(1)		
	m	i	k	t	Çözüm	İşyükü dengesizlik oranı ^c (1)	Ort.		Eniyi	Ortalama dengesizlik oranı ^d (2)
1	5	20	3	20	158.0 ^a	0.1043	158.0	158.0	0,1043	0,0000
2	5	20	5	30	35.2 ^a	0.0140	43.1	35.2	0,0172	0,0032
3	5	30	3	30	163.2 ^a	0.0924	163.2	163.2	0,0924	0,0000
4	5	30	5	40	5.6 ^b	0.0020	32.4	10.4	0,0115	0,0095
5	8	30	3	30	73.5 ^a	0.0175	137.8	126.0	0,0329	0,0153
6	8	30	5	40	80.0 ^b	0.0162	135.9	79.5	0,0278	0,0116
7	8	40	3	40	108.0 ^a	0.0316	163.8	108.5	0,0479	0,0163
8	8	40	5	50	49.0 ^b	0.0059	109.8	70.8	0,0133	0,0074
GAMS çözümlerinden ortalama fark										0.0079

^aVerilen zaman limiti içerisinde (7200 sn) optimal çözüm elde edilmiştir.

^bVerilen zaman limiti içerisinde (7200 sn) tamsayı çözüm elde edilmiştir.

^cGAMS çözümlerinden elde edilen dengesizlik oranları (dengesizlik/toplam işyükü)

^dAlgoritmanın en iyi parametre seti ile çalıştırılmasından elde edilen ortalama dengesizlik oranları

5. SONUÇ (CONCLUSION)

Bu çalışmada, EİS planlama problemlerinden olan parça seçimi ve makina yükleme problemleri sıralı bir karışık tamsayılı programlama modeli ile ifade edilmiş ve çözümleri için uzun dönem hafızalı bir tabu arama algoritması geliştirilmiştir. Algoritmanın en iyi parametre setinin belirlenmesinde 3^k faktöryel tasarım kullanılmıştır. En iyi parametre seti ile test problemleri üzerinde denenilen algoritmanın parça seçimi problemi için hemen her koşulda optimale ulaştığı, makina yükleme problemi için ise GAMS çözümlerinden ortalama binde 7.9'luk bir sapma elde ettiği görülmüştür. Geliştirilen tabu arama algoritması, matematiksel model yardımıyla iki aşamada çözülebilen zor bir problemi tek aşamada kısa sürelerde iyi sonuçlar elde ederek çözmesi açısından oldukça kullanışlıdır.

TEŞEKKÜR (ACKNOWLEDGEMENT)

Bu çalışma, 06/2008-29 kodlu BAP projesi kapsamında gerçekleştirilmiştir.

KAYNAKLAR (REFERENCES)

1. Kusiak, A., "Flexible Manufacturing Systems: A structural approach", **Int J Prod Res**, 23(6), 1057-1073, 1985.
2. Co, C. H., Biermann J. S., Chen S. K., "A methodical approach to the flexible manufacturing system batching, loading and tool configuration", **Int J Prod Res**, 28(12), 2171-2186, 1990.
3. Moreno, A. A., Ding, F. Y., "Heuristics for the FMS loading and part type selection problems", **Int J Flex Manuf Syst**, 5, 287-300, 1993.
4. Nagarjuna, N., Mahesh, O., Rajagopal, K., "A heuristic based on multi-stage programming approach for machine-loading problem in a flexible manufacturing system", **Robot Comput Integrated Manuf**, 22, 342-352, 2006.
5. Goswami, M., tiwari, M. K., "A reallocation-based heuristic to solve a machine loading problem with material handling constraint in a flexible manufacturing system", **Int J Prod Res**, 44(3), 569-588, 2006.
6. Tiwari, M. K., Saha, S., Mukhopadhyay, S. K., "Heuristic solution approaches for combined-job sequencing and machine loading problem in flexible manufacturing systems", **Int J Adv Manuf Technol**, 31, 716-730, 2007.
7. Srivastava, B., Chen, W. H., "Batching in production planning for flexible manufacturing systems", **Int J Prod Econ**, 43, 127-137, 1996.
8. Mukhopadhyay, S. K., Singh, M. K., Srivastava, R., "FMS machine loading: a simulated annealing approach", **Int J Prod Res**, 36(6), 1529-1547, 1998.
9. Arıkan, M., Erol, S., "Meta-heuristic approaches for part selection and tool allocation in flexible manufacturing systems", **Int J Comput Integrated Manuf**, 19(4): 315-325, 2006.
10. Srivastava, B., Chen, W., "Part type selection problem in flexible manufacturing systems: tabu search algorithms", **Ann Oper Res**, 41, 279-297, 1993.
11. Sarma, U. M. B., Kant, S., Rai, R., Tiwari, M. K., "Modelling the machine loading problem of FMSs and its solution using a tabu-search-based heuristic", **Int J Comput Integrated Manuf**, 15(4): 285-295, 2002.
12. Kumar, N., Shanker, K., "A genetic algorithm for FMS part type selection and machine loading", **Int J Prod Res**, 38(16), 3861-3887, 2000.
13. Tiwari, M. K., Vidyarthi, N. K., "Solving machine loading problems in a flexible manufacturing system using a genetic algorithm based heuristic approach", **Int J Prod Res**, 38(14), 3357-3384, 2000.
14. Yang, H., Wu, Z., "GA-based integrated approach to FMS part type selection and machine-loading problem", **Int J Prod Res**, 40(16), 4093-4110, 2002.
15. Kumar, A., Prakash, Tiwari, M. K., Shankar, R., Baveja, A., "Solving machine-loading problem of a flexible manufacturing system with constraint-based genetic algorithm", **Eur J Oper Res**, 175, 1043-1069, 2006.
16. Prakash, A., Khilwani, N., Tiwari, M. K., Cohen, Y., "Modified immune algorithm for job selection and operation allocation problem in flexible manufacturing systems", **Adv Eng Software**, 39, 219-232, 2008.
17. Biswas, S., Mahapatra, S. S., "Modified particle swarm optimization for solving machine-loading problems in flexible manufacturing systems", **Int J Adv Manuf Technol**, 39, 931-942, 2008.
18. Prakash, A., Tiwari, M. K., Shankar, R., "Optimal job sequence determination and operation machine allocation in flexible manufacturing systems: an approach using adaptive hierarchical ant colony algorithm", **J Intell Manuf**, 19, 161-173, 2008.
19. Swarnkar, R., Tiwari, M. K., "Modeling machine loading problem of FMSs and its solution methodology using a tabu search and simulated annealing-based heuristic approach", **Robot Comput Integrat Manuf**, 20, 199-209, 2004.
20. Tabucannon, M. T., Dentcho, N. B., Basu, S., "Using simulation to evaluate the batching approach to part type selection in flexible manufacturing systems", **Integrated Manuf Syst**, 9(1), 5-14, 1998.
21. Liang, M., Dutta, S. P., "An integrated approach to the part selection and machine loading problem in a class of flexible manufacturing systems", **Eur J Oper Res**, 67, 387-404, 1993.
22. Glover, F., "Tabu Search-Part I", **ORSA Journal on Computing**, 1(3), 190-206, 1989.
23. Glover, F., "Tabu Search-Part II", **ORSA Journal on Computing**, 2(1), 4-32, 1990.