ANADOLU ÜNİVERSİTESİ BİLİM VE TEKNOLOJİ DERGİSİ
ANADOLU UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
Cilt/Vol.:3 - Sayı/No: 1 : 77-88 (2002)

ARAŞTIRMA MAKALESİ/**RESEARCH ARTICLE**

TESİS YERLEŞİM PROBLEMİ İÇİN BİR BULANIK-TABU ARAMA YAKLAŞIMI

Orhan TÜRKBEY[1,2], Çiğdem ALABAŞ[1,3]

ÖZ

Üretim sistemlerindeki işçilik maliyetinin yaklaşık %20-50'sini malzeme taşıma maliyeti oluşturabilmektedir. Bu nedenle tesis planlaması, kaynakların üzerinde en çok durduğu konulardan birisi olmuştur. Ayrıca bu problemin kombinatoryal yapısı ve NP-tam doğası, hala artan bir ilgi alanı olmasına yol açmaktadır. Bu çalışmada, kareli atama problemi olarak modellenen tesis düzenlemesi için zeki sezgisel teknikler arasında olan tabu arama algoritması geliştirilmiştir. Geliştirilen tabu arama algoritmasındaki tabu sürelerinin belirlenmesi için, kaynaklarda görülen geleneksel yaklaşımlardan farklı olarak bulanık küme kuramından yararlanılmaktadır. Önerilen algoritma, sabit tabu süresini kullanan klasik tabu arama algoritması ve en temel sezgisel teknik olan rassal arama algoritması ile rassal olarak üretilen test problemleri üzerinde karşılaştırılmıştır. Bu karşılaştırmada başarı ölçütü olarak, algoritmaların buldukları çözümlerin kalitesi ve aradıkları çözüm sayıları dikkate alınmış ve sonuçta geliştirilen tabu arama algoritmasının daha iyi bir başarıya sahip olduğu gözlenmiştir.

Anahtar Kelimeler: *Kareli atama problemi, Bulanık küme kuramı, Tabu arama algoritması, Tesis düzenleme, Sezgisel algoritma.*

A FUZZY-TABU SEARCH APPROACH FOR THE PLANT LAYOUT PROBLEM

ABSTRACT

Material handling costs could reach up to 20 to 50 percent of wages in a production system. So layout planning has received a considerable amount of attention in the research literature. Additionally, its combinatorial structure and NP-complete nature, increased this interest. A tabu search algorithm as a smart heuristic technique is developed by this study to solve a type of layout problem, formulated as a quadratic assignment model. Constrasting from convetional approaches found in the literature, this study made use of fuzzy set theory to determine the size of tabu list (or tabu tenure). Proposed algorithm is compared with classical tabu search algorithm and random search method on randomly generated test problems, from point of view of solution quality and number of search point criteria and a better performance is obtained.

Key Words: *Quadratic assignment problem, Fuzzy set theory, Tabu search algorithm, Plant layout, Heuristic algorithm.*

1.GİRİŞ

Üretim sistemlerinin başarısında tesis yerleşim probleminin stratejik bir konumu olmasının başlıca nedeni, işçilik maliyetinin %20-%50'nin malzeme taşıma (yönetim) giderlerinden kaynaklanmasıdır. Tesis yerleşimi problemi için iyi bir çözüm, sistemin genel verimliliğine katkıda bulunacaktır. Buna karşın kötü bir yerleşim, ara stokların birikmesine, malzeme taşıma sistemlerinin çakışmasına, hazırlık zamanlarının artmasına ve sistemde uzun kuyrukların oluşmasına yol açacaktır (Jajodia vd., 1992). Tesis yerleşimi problemi, uzun dönemde yüksek maliyetli yatırım gerektirebilir. Mevcut bir tesisin yeniden yerleştirilmesi ya da iyileştirilmesi ise pahalı ve zaman alıcı olacağı gibi işçi faaliyetlerini ve malzeme akışını da engeller (Sule, 1988). Ancak, tesis yerleşimi tasarımının stratejik önemi göz ardı edilmemelidir.

1 Endüstri Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Gazi Üniversitesi, Maltepe, 06570, Ankara, Türkiye.
2 **E-posta:** turkbey@mmf.gazi.edu.tr
3 **E-posta:** alabas@mmf.gazi.edu.tr
Geliş: 21 Aralık 2000; **Düzeltme:** 30 Nisan 2001; **Kabul:** 25 Eylül 2001.

Tesis yerleşimi problemlerinin çözülmesi için kullanılan en yaygın model, kareli atama (*quadratic assignment*) problemidir. Bu amaçla kullanılan diğer modeller ise, kareli küme kapsama (*quadratic set covering*), doğrusal tamsayılı programlama (*linear integer programming*), karışık tamsayılı programlama (*mixed integer programming*) ve çizge kuramı modelleri (*graph theoretic models*) olarak sıralanabilir (Alvarenga vd., 2000).

Tesis yerleşimi problemini, Kareli Atama Problemi (KAP) olarak ilk kez Koopmans ve Beckman (1957) modellemişlerdir. Bu probleme "kareli atama" isminin verilme nedeni, amaç fonksiyonu değişkenlerinin ikinci dereceden polinom bir fonksiyon olması ve kısıtların da doğrusal atama probleminin kısıtlarına benzemesidir. KAP'nin amacı, akış ve dolaşılan uzaklığın çarpımı olarak ifade edilen malzeme taşıma maliyetini en küçükleyecek şekilde, n sayıda tesisin (makinanın, bölümün veya iş istasyonunun) n sayıda yere eniyi atamasını belirlemektir. KAP modeli aşağıda verilmektedir.

$$\begin{aligned}
\text{k.a} \quad & \sum_{j=1}^{n} x_{ij} = 1, \quad \forall i \\
& \sum_{i=1}^{n} x_{ij} = 1, \quad \forall j \qquad\qquad (1) \\
& x_{ij} = 0 \text{ veya } 1. \\
\text{Enk} \quad & \sum_{i=1}^{n}\sum_{p=1}^{n}\sum_{j=1}^{n}\sum_{q=1}^{n} f_{ip}\, d_{jq}\, x_{ij}\, x_{pq}
\end{aligned}$$

Yukarıdaki modelde x_{ij} karar değişkeni, eğer i tesisi j alanına atanırsa 1 değerini, aksi halde 0 değerini almaktadır. Tesis i ve p arasındaki iş akışı f_{ip} ile, j ve q alanları arasındaki uzaklık ise d_{jq} ile ifade edilmektedir. Bu problemde amaç fonksiyonu karelidir ve dışbükey olmayan bir yapıya sahiptir. KAP için birden çok yerel eniyi çözüm vardır ve çözüm uzayında n! kadar çözüm noktası bulunmaktadır.

2. KARELİ ATAMA PROBLEMLERİ İÇİN KAYNAK İNCELEMESİ

KAP, NP-tam (NP-*complete*) bir problemdir. Gilmore (1963), Lawler (1963), Burkard ve Stratmann (1978), Bazaraa ve Elshafei (1979) ve Bazaraa ve Sherali (1980) bu problem için en iyi sonuçları veren yöntemler geliştirmişlerdir. Ancak en iyi yöntemlerin hesaplama yeteneği, en fazla 20 tesisli yerleşim problemlerini çözebilmelerine izin vermektedir (Resende vd., 1995).

Tesis yerleşimi probleminin kombinatoryal yapısı ve eniyileme algoritmalarının büyük boyutlu problemleri çözmedeki yetersizliği, araştırmacıları bu konuyla ilgili sezgisel algoritmalar geliştirmeye yönlendirmiştir. Söz konusu sezgisel yöntemler, yapılandırma (*construction*) ve iyileştirme (*improvement*) yöntemleri olmak üzere iki sınıfa ayrılmaktadır. Yapılandırma yöntemleri, boş bir alanla işleme başlamakta ve düzenleme tamamlanana kadar ardıl olarak tesisleri seçerek yerleştirmektedir. Diğer taraftan iyileştirme metotlarında işlem, mevcut bir düzenleme üzerinde başlar. Bu başlangıç düzenlemesi bir yapılandırma algoritmasının sonucu olabileceği gibi rassal olarak da seçilebilir. İyileştirme metotları mevcut düzenlemeden hareketle, tesis birimlerinin yerlerini karşılıklı değiştirerek daha iyi yerleşim tasarımlarına ulaşmaya çalışır. Foulds (1983), Kusiak (1990) ve Burkard (1990), KAP için sezgisel algoritmalar konusunda ayrıntılı birer kaynak incelemesi sunmuşlardır.

Zeki sezgisel teknikler olan genetik algoritmalar (*genetic algorithms*), tavlama benzetimi (*simulated annealing*), tabu arama (*tabu search*) ve yapay sinir ağlarının (*artificial neural networks*) geliştirildikleri zamandan günümüze kadar pek çok probleme uygulanmaları ve bu problemler için oldukça iyi çözümler vermeleri, araştırmacıların son zamanlarda, KAP için zeki sezgisel yaklaşımlar kullanmalarına yol açmıştır.

Kirkpatrick vd. (1983) ve Cerny (1985) tarafından birbirlerinden bağımsız olarak öne sürülen "tavlama benzetimi" fiziksel tavlama sürecini taklit eden olasılıklı bir metotdur ve bir çok araştırmacı farklı tesis yerleşimi problemlerini çözmek için bu yöntemi kullanmışlardır. Kim ve Kim (1998), Chwif vd. (1998), Wang vd. (1998), Bazargan ve Kaebernick (1997), Meller ve Bozer (1996), Yip ve Pao (1994) tesis yerleşimi problemlerine tavlama benzetimi yaklaşımını en yeni uygulayan araştırmacılardır.

Zeki sezgisel tekniklerden bir diğeri olan genetik algoritmalar, doğanın "en iyi olan hayatta kalır" prensibini taklit etmektedirler. Genetik algoritmaların öncüleri Holland (1975), DeJong (1975) ve Goldberg (1989)'dir. Genetik algoritmalar günümüze kadar çeşitli tesis yerleşimi problemlerini çözmek amacıyla kullanılmışlardır. Zhang vd. (2000), Al-Hakim (2000), Gau ve Meller (1999), Kochhar ve Heragu (1999), Hamamoto vd. (1999), Tam ve Chan (1998), Mak vd. (1998), Suresh vd. (1995) genetik algoritmaları son zamanlarda tesis yerleşimi problemlerine başarıyla uygulamışlardır.

Bu günkü modern şekli Glover (1989, 1990) tarafından verilen "tabu arama", yerel eniyilerden kurtulmak amacıyla yerel arama usullerine (prosedürlerine) yol gösteren zeki bir sezgisel tekniktir. Tesis yerleşimi problemlerine uygulanmış tabu aramaya dayalı en yeni çalışmalara, Abninnour-Helm ve Hadley (2000), Alvarenga vd. (2000), Chittratanawat ve Noble (1999), Chiang ve Chiang (1998), Chiang ve Kouvelis (1996) örnek olarak gösterilebilir.

Öncülüğü McCulloch ve Pitts (1943) ve Rosenblatt (1962)'in çalışmalarından gelen "yapay sinir ağları" insan beyninin çalışma ilkelerini taklit etmektedirler. Rossin vd. (1999), Chung (1999), Kazurhiro vd. (1996), Liao (1994), tesis yerleşimi problemine yapay sinir ağı yaklaşımını uygulayan araştırmacılardır. Diğer taraftan, Badiru ve Arif (1996) ve Raoot ve Rakshit (1994) tesis yerleşimi problemine bulanık küme (Zadeh, 1965) esaslı yaklaşımları uygulayan araştırmacılardır.

3. TABU ARAMA VE BULANIK KÜME KURAMI

Tabu arama yönteminin felsefesi ilk olarak Glover ve Greenberg (1989) ve Hansen (1986) tarafından birbirlerinden bağımsız olarak ileri sürülmüştür. Ancak yönteme bu ismi veren ve onu bugünkü modern şekline getiren araştırmacı Glover (1989, 1990) olmuştur. Tabu arama, "yerel eniyilerden kurtulmak için yöre arama (*neighborhood search*) yöntemlerine yol gösterir".

Tabu arama tekniği, çözüm uzayındaki arama işlemine herhangi bir başlangıç çözümüyle başlar. Bu çözüm üzerinde yapılan küçük değişikliklerle elde edilen komşu çözümler mevcut çözümün yöresini oluşturmaktadır ve bir çözümü başka bir çözüme dönüştürme işlemine "hareket etme" ismi verilmektedir. Klasik yöre arama yöntemleri, mevcut çözümün yöresi içindeki ve mevcut çözümden daha iyi amaç değerine sahip olan bir başka çözüme hareket ederek bu çözümü yeni mevcut çözüm olarak seçerler. Bu işlem yöre içindeki tüm komşu çözümler daha iyi bir amaç değeri vermeyinceye kadar tekrar edilir ve sonuçta bulunan çözüm yerel bir eniyi nokta olur.

Tabu arama tekniği, yerel arama usullerinin bu yerel tuzaklardan kurtulmaları için çeşitli yöntemler kullanır. Öncelikle, mevcut çözümün yöresi içindeki en iyi çözüm, mevcut çözümden daha kötü olsa bile yeni mevcut çözüm olarak seçilmektedir. Ancak bu seçme işlemine, yasaklanan hareketlerle elde edilen komşu çözümler dahil edilmemektedir. Tabu arama, daha önce gerçekleştirilmiş bir hareketin tersinin yapılmasını önlemek amacıyla tabu listesini veya listelerini kullanır. Bunun için mevcut çözümden komşu bir çözüme hareket edildiği zaman bu hareketi eski haline getiren bir başka hareket tabu listesine kaydedilerek yasaklanmakta yani tabu yapılmaktadır. Tabu listesine kaydedilen hareketler belli bir tekrar süresi (tabu süresi) boyunca bu listede kalmakta ve daha sonra tabu durumları kaldırılarak yapılmalarına izin verilmektedir. Tabu aramanın "kısa dönemli hafızasını" oluşturan bu mekanizma, arama boyunca çevrimlerin oluşmasını önlemekte ve aramanın yönünü değiştirerek yerel eniyi tuzaklarından kurtulmayı sağlamaktadır. Arama işlemi, mevcut çözümün yöresi içinden, tabu olmayan ve en iyi amaç değerine sahip komşu çözümün yeni mevcut çözüm olarak seçilmesiyle devam etmektedir. Bu işlemler, maksimum tekrar sayısı gibi belli bir durdurma koşulu gerçekleşene kadar yinelemeli olarak tekrarlanmaktadır. Ayrıca, tabu arama bazı özel durumlarda tabu olmalarına rağmen bazı hareketlerin yapılmalarına izin vermektedir. "Tabu yıkma ölçütleri" tarafından belirlenen bu özel durumlar içerisinde en yaygın kullanılan prensip, tabu bir hareketle elde edilen çözümün arama boyunca o ana kadar bulunmuş en iyi çözümden bile daha iyi bir amaç değerine sahip olması durumunda yapılmasına izin verilmesidir.

Diğer taraftan tabu aramanın uzun dönemli hafızasında, gerçekleştirilen hareketlere ait bazı özellikler tutulmaktadır. Uzun dönemli hafızada genellikle, hareketlerin kalıcılık ve geçicilik sıklıkları yani mevcut çözümü oluşturmak üzere seçildikleri tekrar sayısı ve mevcut çözümden çıkarıldıkları tekrar sayısı kaydedilmektedir. Bu sıklık bilgileri, aramanın belli bölgelere odaklandırılarak "kuvvetlendirilmesi" ve/veya aramanın çözüm uzayındaki farklı bölgelere yönlendirilerek "çeşitlendirilmesi" amacıyla kullanılmaktadır. Etkin bir tabu arama algoritmasında, dikkate alınan kuvvetlendirme ve çeşitlendirme stratejilerinin önemi büyüktür. Tabu arama, ileri sürüldüğü günden günümüze kadar bir çok problemlere uygulanmış ve oldukça iyi sonuçlar elde edilmiştir. Glover ve Laguna çalışmalarında (1993), bu yöntemin uygulama alanları ile ilgili olarak bir kaynak incelemesini vermektedir.

Bulanık kümeler kuramı ilk kez 1960'lı yılların ortasında ileri sürülmüştür. Zadeh (1965) ise bu kuramın temellerini kurmuştur. Bulanık kümeler kuramı, istatistiksel yapıda olmayan belirsizliklerle başa çıkmak amacıyla geliştirilmiştir. Bulanık kümeleri klasik kümelerden ayıran özelliği, elemanlarının üyelik dereceleridir. Klasik kümede bir eleman, kümenin ya üyesidir ya da değildir, dolayısıyla üyelik fonksiyonu bir (0, 1) ikili fonksiyondur. Bulanık kümede ise üyelik fonksiyonu, 0 ve 1 arasında herhangi bir değeri alabilmektedir. Bu durumda kesirli bir üyelik derecesi, o elemanın kümeye kısmi üyeliğini ifade etmektedir. Bu özellik, kümenin ana hatlarını kesin olarak tanımlayan sınırlar olmadığında çok kullanışlı olmaktadır. Bulanık üyelik fonksiyonun esas avantajlarından biri, kümenin bir üyesi olma durumundan, üyesi olmama durumuna dereceli olarak geçiş imkanının tanınmasıdır (Hassanein ve Cherlopalle, 1999).

Bu çalışmada tabu aramayı ve bulanık kümeler kuramını birlikte düşünmeyi gerektiren neden, tabu aramadaki "sistematik yol gösterme" varsayımıdır. Tabu aramada kullanılan kuvvetlendirme ve çeşitlendirme stratejileri, iyi çözüm bölgelerine yoğunlaşma ve farklı çözüm bölgelerine ulaşma esaslarına dayanır. Ancak bu iki strateji arasındaki denge, algoritmanın başarısı açısından çok önemlidir. Dolayısıyla, "arama, iyi çözüm bölgelerine ne kadar yoğunlaşmalı ve ne zaman farklı çözüm bölgelerine yönlenmeli ?" sorularına önceden verilebilecek kesin cevaplar yoktur. Esasında bu sorular, iç etkileşimlidirler ve cevapları da aramanın seyrine bağlı olarak değişmektedir. Bu nedenle çalışmada tabu arama yönteminin, önceden tanımlanmış bir bulanık üyelik fonksiyonunu kullanarak, arama esnasında söz konusu sorulara cevap bulmasını sağlayacak yeni bir bulanık-tabu arama algoritması geliştirilmiştir.

4. KARELİ ATAMA PROBLEMİ İÇİN BULANIK-TABU ARAMA ALGORİTMASI

Bu çalışmada, KAP olarak modellenen tesis yerleşimi problemi için tabu aramaya dayalı sezgisel bir algoritma geliştirilmiştir. Bu algoritma, içerisinde bulanık küme ilkelerini barındırması açısından kaynakta görülen tabu arama uygulamalarından tamamıyla farklıdır. Geliştirilen Bulanık-Tabu Arama (BTA) algoritmasının kullandığı yöre yapısı, tabu listesinin yönetimi ve bulanık tabu sürelerinin belirlenmesi izleyen bölümlerde sunulmuştur.

4.1. Yöre Yapısı

BTA algoritmasında, çözüm uzayındaki herhangi bir çözüm vektörü hangi alana hangi makinanın (veya tesisin, birimin) atandığını göstermektedir. BTA algoritmasında kullanılan kodlama yapısına göre makine numaralarının sırası, bu makinaların hangi alanlara yerleştirildiğini ifade etmektedir. Örneğin 5 boyutunda bir tesis yerleşimi problemi için $x = \{2, 1, 3, 5, 4\}$ çözüm vektörü, 1 makinasının 2 numaralı alana, 2 makinasının 1 numaralı alana, 3 makinasının 3 numaralı alana, 5 makinasının 4 numaralı alana ve 4 makinasının 5 numaralı alana yerleştirildiğini göstermektedir. Burada bütün makinaların boyutlarının her alana yerleştirilmeye uygun olduğu varsayılmaktadır.

Kullanılan yöre yapısına göre, mevcut bir çözümün komşuları, bu çözüm üzerinde yapılabilecek mümkün iç-değişim (*exchange*) hareketleriyle elde edilmektedir, yani seçilen makinaların karşılıklı olarak yerleri değiştirilmektedir. Örnek olarak x vektörünün iç-değişim hareketiyle elde edilen komşu çözümleri Tablo 1'de verilmektedir.

İç-değişim hareket mekanizmasına göre, n problem boyutunu (alan/makina sayısını) göstermek üzere, herhangi bir çözümün $n(n-1)/2$ tane komşu çözümü elde edilebilmektedir. İterasyonlu bir yöntem olan BTA algoritması, her tekrarda mevcut çözümün $n(n-1)/2$ tane komşu çözümünden oluşan yöresini araştırmaktadır. Ayrıca, büyük boyutlu problemler için ($n > 20$) yöre genişliğinin komşu çözümlerin bir altkümesine sınırlandırılması da mümkündür.

4.2. Tabu Listesinin Yönetimi

BTA algoritmasında daha önce gerçekleştirmiş bir iç-değişim hareketini bozan hareketler, yani mevcut çözümü eski haline döndüren hareketler tabu_süre ile gösterilen belli bir tekrar sayısı boyunca yasaklanmakta (tabu sınıfına alınmakta) ve böylece çözüm uzayındaki aramanın yönü değiştirilerek farklı çözüm bölgelerine ulaşma imkanı sağlanmaktadır. Bu ilke, BTA algoritmasının yerel eniyi tuzaklarından kurtulmasına imkan tanımaktadır. Bu amaçla, *tabu_liste[i, j]* ($i = 1, 2, ..., n, j = 1, 2, ..., n$) isimli bir tabu listesi tutulmaktadır.

Tabu_liste[i, j] listesinin yönetimi kısaca şöyle açıklanabilir: Mevcut çözüm $x[k]$ ($k = 1, 2, ..., n$) makinelerin herhangi bir sıralamasını (alanlara yerleştirilme düzenini) göstersin ve bu sıralamada p. alandaki a makinasıyla, q. alandaki b makinasının iç-değişimi ile elde edilen bir komşu çözüm yeni mevcut çözüm olarak kabul edilmiş olsun. Bu durumda *tabu_liste[i, j]* listesi, a makinasının yeni atandığı q alanından ve b makinasının yeni atandığı p alanından ayrılmasının yasaklanması amacıyla, *iter* mevcut tekrar sayısını göstermek üzere aşağıdaki gibi güncelleştirilmektedir:

(*tabu_liste[a, q] = iter*) ve (*tabu_liste[b, p] = iter*)

Tabu_liste[i,j] listesi bir iç-değişim hareketinin tabu olmaya başladığı ilk iterasyonun değerini tutmaktadır ve takip eden *tabu_süre* kadarki tekrar sayısı boyunca söz konusu hareket tabu olmaktadır. Böylece algoritmanın sonraki tekrarlarında r. alandaki c makinasıyla, s. alandaki d makinasının yer değiştirmesi aşağıdaki eşitsizliklerden birisi sağlandığında tabu olmaktadır:

Tablo 1. $x = \{2, 1, 3, 5, 4\}$ Çözümünün İç-Değişim Hareketiyle Elde Edilen Komşu Çözümleri.

	İç-değişimler	1	2	3	4
1	2. makinanın yer değiştirmesiyle elde edilen komşular	$x'_1 = \{1, 2, 3, 5, 4\}$	$x'_2 = \{3, 1, 2, 5, 4\}$	$x'_3 = \{5, 1, 3, 2, 4\}$	$x'_4 = \{4, 1, 3, 5, 2\}$
2	1. makinanın yer değiştirmesiyle elde edilen komşular	$x'_5 = \{2, 3, 1, 5, 4\}$	$x'_6 = \{2, 5, 3, 1, 4\}$	$x'_7 = \{2, 4, 3, 5, 1\}$	(1,1) ile aynı
3	3. makinanın yer değiştirmesiyle elde edilen komşular	$x'_8 = \{2, 1, 5, 3, 4\}$	$x'_9 = \{2, 1, 4, 5, 3\}$	(1,2) ile aynı	(2,1) ile aynı
4	5. makinanın yer değiştirmesiyle elde edilen komşular	$x'_{10} = \{2, 1, 3, 4, 5\}$	(1,4) ile aynı	(2,3) ile aynı	(3,1) ile aynı

($iter \leq tabu\_liste[c, r] + tabu\_süre$) veya ($iter \leq tabu\_liste[d, s] + tabu\_süre$)

Tabu_liste[i, j] listesinin yönetimi küçük bir örnek üzerinde açıklanacak olursa; mevcut çözüm x[k], $x[1] = 2$, $x[2] = 1$, $x[3] = 3$, $x[4] = 5$, $x[5] = 4$ sıralamasını göstersin (yani, $x = \{2, 1, 3, 5, 4\}$ olsun) ve birinci tekrarda 2. ve 4. makinaların yer değiştirmesiyle elde edilen komşu çözüm yeni mevcut çözüm olarak kabul edilmiş olsun. Bu durumda aşağıdaki atamalar gerçekleştirilir:

$Tabu\_liste[2, 5] = 1$

$Tabu\_liste[4, 1] = 1$

$x[1] = 4, x[5] = 2$

İkinci tekrarda ise, mevcut çözümde 2. ve 5. makinaların iç-değişimi ile elde edilen bir komşu çözüm, "($2 \leq tabu\_liste[2, 5] + tabu\_süre$) veya ($2 \leq tabu\_liste[5, 4] + tabu\_süre$)" eşitsizliklerinin ikisinden birisi sağlanıyorsa tabu olmaktadır. Burada, $tabu\_liste[2, 5] = 1$ olduğundan dolayı, tabu süresinin 1 ya da daha büyük bir değer alması söz konusu komşu çözümün tabu olmasına yol açacaktır. BTA algoritmasında tabu süresinin hangi değeri alacağı seçilen bir bulanık üyelik fonksiyonuna göre belirlenmektedir. Bir sonraki bölümde bu konu üzerinde durulacaktır.

4.3. Bulanık Tabu Süresi

Günümüze kadar tabu arama algoritmasına dayalı olarak yapılan çalışmalar, tabu süresinin algoritmanın başarısı üzerinde oldukça etkili olduğunu göstermiştir. Tabu süresinin kısa seçilmesi, aramanın çözüm uzayında belli bölgelere odaklanmasına (yani bölgesel olarak kuvvetlendirilmesine), tabu süresinin uzun seçilmesi ise, aramanın çözüm uzayındaki farklı bölgelere yönlenmesine (yani genel olarak çeşitlendirilmesine) yol açmaktadır. Ancak tabu süresi gereğinden kısa seçilirse, belli sayıda tekrar sonunda aynı çözümlere ulaşılacak dolayısıyla çevrimler oluşacak aksine tabu süresi fazla uzun seçilirse çözüm uzayında kötü bölgelerden çıkılamayarak çözüm kalitesi düşecektir. Her iki halde de elde edilen çözümler bütünsel eniyiden çok uzak olacaktır. Bu nedenle kaynakta, sabit tabu sürelerinin yerine dinamik tabu sürelerini kullanan çalışmalara rastlanmaktadır (Glover ve Laguna, 1997). Bunun yanı sıra, tabu arama algoritmasında kuvvetlendirme ve çeşitlendirme stratejileri birbirlerinden ayrı düşünülmemektedir. Her iki stratejinin eşzamanlı olarak kullanılması ve birbirlerini dengelemesi algoritmanın başarısı açısından oldukça önemlidir.

Bu çalışmada ise, tabu süresinin belirlenmesi için farklı bir yaklaşım geliştirilmiştir. Bu yaklaşımın temel fikri, "arama boyunca seyrek olarak tekrar eden hareketler için tabu süresini kısa seçerek ve sık olarak tekrar eden hareketler için tabu süresini uzun seçerek, arama yönünün bir taraftan çeşitlendirilirken diğer taraftan kuvvetlendirilmesini sağlamaktır". Ancak bu noktada, "ne kadar seyrek?" ya da "ne kadar sık?" sorularına karşı verilebilecek göreceli cevaplar, bulanık küme kuramından yararlanma gerekliliğini ortaya koymuştur. Yapılan denemeler sonucunda, makinaların alanlara atanma sıklığı ile tabu süresi arasında aynı yönlü bir doğrusal ilişki kuran bulanık üyelik fonksiyonunun ilgilenilen tesis yerleşimi problemi için uygun olduğu tespit edilmiştir. *Tabu_süresi_alt*, tabu süresinin alt sınırını, *tabu_süresi_üst*, tabu süresinin üst sınırını ve $\mu_{i,j}$, i makinasının j alanına atanmasının 0 ile 1 arasında normalize edilen sıklığını gösteren üyelik değeri olmak üzere, seçilen bulanık üyelik fonksiyonu (2) eşitliği ile gösterilmektedir. Şekil 1'de ise söz konusu bulanık üyelik fonksiyonunun grafik gösterimi verilmektedir.

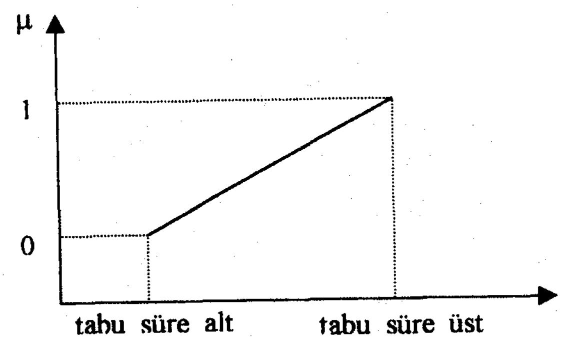
$$tabu\_süre_{i,j} = tabu\_süre\_alt - \mu_{i,j}\,(tabu\_süre\_alt - tabu\_süre\_üst) \quad (2)$$

Buna göre, i makinasını j alanına atayan bir iç-değişim hareketinin tabu süresi (2) eşitliğinden hesaplanmaktadır. $\mu_{i,j}$ üyelik değerinin belirlenmesi için ise, her tekrarda gerçekleştirilen iç-değişim hareketiyle gerçekleşen atamalar için bir $sıklık[i, j]$ (i = 1, ..., n, j = 1, ..., n) listesi tutulmaktadır. Bir çeşit sayaç görevi gören bu liste, mevcut çözüm üzerinde p. alandaki a makinasıyla, q. alandaki b makinasının yer değiştirmesiyle yeni mevcut çözümü oluşturan bir hareketten sonra aşağıdaki gibi güncelleştirilmektedir.

$sıklık[a, q] = sıklık[a, q] + 1$

$sıklık[b, p] = sıklık[b, p] + 1$

$Sıklık[i, j]$ listesi daha sonra 0 ile 1 arasında normalize edilerek $\mu_{i,j}$ üyelik fonksiyonu değerlerine dönüştürülmektedir. Bu normalizasyon işlemi için (3) eşitliğinde verilen denklem kullanılmaktadır.



Şekil 1. Tabu Sürelerinin Belirlenmesi İçin Bulanık Üyelik Fonksiyonu.

$$\mu_{i,j} = \frac{sıklık\ [i,j] - min(sıklık[i,j]}{maks(sıklık\ [i,j]) - min(sıklık[i,j]} \quad (3)$$

4.4. BTA Algoritmasının Adımları

Yinelemeli bir algoritma olan BTA algoritması, rassal olarak seçilen bir başlangıç çözümünü mevcut çözüm olarak dikkate alıp aramaya başlamakta ve mevcut çözümün yöresi içinde bulunan komşu çözümleri Bölüm 4.1'de açıklanan iç-değişim hareket mekanizmasına göre tespit etmektedir. Bu komşular içerisinden en iyi amaç fonksiyonu değerine yani en düşük malzeme taşıma maliyetine sahip olan; aynı zamanda, Bölüm 4.2'de verilen tabu listesi yönetimine göre, Bölüm 4.3'de tanımlanan bulanık tabu süresini dikkate alarak tabu olmayan komşu çözüm yeni mevcut çözüm olarak seçilmektedir. Ancak tabu yıkma ölçütü göz önüne alındığında, eğer tabu olarak sınıflandırılmış bir iç-değişim hareketiyle elde edilen komşu çözümün amaç fonksiyonu değeri, arama boyunca o ana kadar bulunmuş en iyi amaç fonksiyonu değerinden daha iyi ise, tabu olmasına rağmen söz konusu komşu çözümün yeni mevcut çözüm olarak kabul edilmesine izin verilmektedir. Yeni mevcut çözümün seçilmesinden sonra *tabu_liste*[*i, j*] ve *sıklık*[*i, j*] (i=1, ..., n, j=1 ,..., n) listeleri güncelleştirilmektedir. Bütün bu işlemler, önceden belirlenen maksimum tekrar sayısına ulaşılana kadar tekrarlanmaktadır. BTA algoritmasının adımları, f amaç fonksiyonunu göstermek üzere aşağıda verilmektedir.

Adım 1. Rassal olarak bir x başlangıç çözümü seç ve onu mevcut çözüm olarak ata:

$x_{mev} = x$

En iyi çözüm $x_{eniyi} = x_{mev}$ olsun.

İterasyon sayacına başlangıç değerini ver: *iter* = 1

Boş tabu ve sıklık listeleri ile aramaya başla.

Adım 2. İç-değişim hareket mekanizmasına göre x_{mev} çözümünün komşularını bul:

$x_{kom}^{(k)}$, k = 1, ..., $n(n\text{-}1)/2$

Adım 3. k = 1, ..., $n(n\text{-}1)/2$ için $f(x_{kom}^{(k)})$ değerlerini hesapla. En iyi $f(x_{kom}^{(k)})$ değerine sahip ve tabu olmayan ya da f(xkom(k)) < f(xeniyi) ölçütünü sağlayan komşu çözüm xkom(s) olsun.

Adım 4. Mevcut çözümü güncelleştir: $x_{mev} = x_{kom}^{(s)}$

Eğer $f(x_{mev}) < f(x_{eniyi})$ ise $x_{eniyi} = x_{mev}$ olsun.

Adım 5. Tabu listesini ve sıklık listesini güncelleştir.

Sıklık listesini normalize ederek üyelik değerlerine dönüştür.

Adım 6. *iter = iter* + 1

Eğer iter>maksimum_iter ise dur, değilse adım 2'ye dön.

5. DENEYSEL ÇALIŞMA

Geliştirilen BTA algoritmasının etkinliğini göstermek amacıyla bu algoritma, statik tabu süresini kullanan Klasik Tabu Arama (KTA) algoritmasıyla ve Rassal Arama (RA) algoritmasıyla karşılaştırılmıştır. KTA algoritması, Bölüm 4'de açıklanan BTA algoritmasıyla tamamıyla aynı prensipleri dikkate almaktadır. Aralarındaki tek fark, BTA algoritmasında tabu süreleri bulanık üyelik fonksiyonuna göre belirlenirken, KTA algoritmasında sabit bir tabu süresinin kullanılmasıdır. BTA ve KTA algoritmalarını karşılaştırmadaki temel amaç, bulanık tabu süresi kullanmanın, tabu arama metodunun kareli atama problemi üzerindeki etkinliğini artırıp artırmayacağını belirlemektir. Diğer taraftan en basit sezgisel arama metodu olan RA algoritmasında, çözüm uzayından raassal olarak belli bir genişlikte örnekleme yapılmakta ve bu çözümler arasından en düşük amaç fonksiyonu değerine sahip olan çözüm ilgili denemenin sonucu olarak dikkate alınmaktadır. RA algoritmasının tek parametresi olan örnekleme genişliği, aynı temelde karşılaştırmayı sağlayabilmek amacıyla BTA ve KTA algoritmalarının çözüm uzayında aradıkları ortalama çözüm sayısına eşit alınmıştır.

Algoritmaların başarılarını karşılaştırmak amacıyla küçük ve büyük boyutlu olmak üzere rassal olarak KAP'leri üretilmiştir. 5, 6, 7, 8 ve 9 boyutunda küçük problemler için sayımlama (*enumeration*) yöntemiyle eniyi çözümler bulunmuştur. Bu küçük boyutlu problemlerin bazıları için tesisler arası uzaklık ve akış matrisleri ek 1'de verilmektedir. 10, 20 ve 30 boyutunda problemler için ise eniyi çözümler bilinmemektedir. Ayrıca her boyutta 5'er farklı KAP (farklı uzaklık ve iş akışı matrisleri) bulunmaktadır. Buna göre toplam 8x5 = 40 KAP üzerinde BTA, KTA ve RA algoritmaları karşılaştırılmaktadır. Ancak karşılaştırmaların yapılmadan önce KTA ve BTA algoritmalarının parametre değerlerinin belirlenmesi amacıyla, seçilen farklı parametre değerleri için algoritmaların en yüksek başarıya sahip olduğu durum aranmıştır. Yapılan ön denemeler KTA ve BTA algoritmalarının parametrelerinin problem boyutuna (n) bağlı olduğunda daha iyi başarıya sahip olduklarını göstermiştir. KTA algoritmasında *tabu_süre* tek parametre olarak kullanılmaktadır. BTA algoritmasında kullanılan parametreler ise, (2) eşitliği ile verilen bulanık tabu süresi fonksiyonundaki tabu süresinin alt ve üst sınırlarıdır (*tabu_süre_alt, tabu_süre_üst*). Tablo 2'de KTA algoritmasının ve Tablo 3'de BTA algoritmasının, problem boyutuna bağlı olarak belirlenen farklı parametre seviyelerinde ortalama başarıları verilmektedir. Söz konusu başarılar için, elde edilebilen ortalama amaç değeri ölçüt olarak dikkate alınmaktadır.

Tablo 2. KTA Algoritmasının Farklı Parametre Seviyelerindeki Ortalama Başarısı.

N	*tabu_süre*			
	$n/3$	$n/2$	n	$2n$
5	15602,6	15602,6	15557,0	15557,0
6	16149,8	16149,8	15978,0	15978,0
7	25070,8	25021,0	24671,2	24613,0
8	29222,0	29158,4	28518,0	28518,0
9	43107,8	42969,0	42833,8	42695,0
10	43719,2	43719,2	43719,2	43470,0
20	702533,0	698096,8	692972,0	694188,2
30	480507,8	480785,2	479428,6	480155,6

Tablo 3. BTA Algoritmasının Farklı Parametre Seviyelerindeki Ortalama Başarısı.

N	*tabu_süre_alt – tabu_süre_üst*				
	$n/3 - n/2$	$n/3 - n$	$n/2 - n$	$n/2 - 2n$	$n - 2n$
5	15602,6	15557,0	15557,0	15557,0	15557,0
5	16107,2	15978,0	15978,0	15978,0	15978,0
7	24826,4	24807,6	24807,6	24613,0	24613,0
8	29158,4	28654,8	28881,2	28518,0	28518,0
9	42969,0	42833,8	42833,8	42695,0	42695,0
10	43719,2	43533,8	43702,4	43470,0	43470,0
20	697941,2	693958,2	694002,8	692881,6	694495,8
30	479572,2	479690,6	479557,2	480604,0	481149,6

Tablo 2 incelendiğinde 5 – 10 boyutunda olan problemler için KTA algoritmasında kullanılan tabu süresinin problem boyutunun iki katı seçilmesinin en iyi sonucu verdiği görülmektedir. 20 ve 30 boyutunda olan problemler için ise tabu süresi problem boyutuna eşit olarak dikkate alındığında KTA algoritmasının başarısı daha yüksek olmaktadır.

BTA algoritmasının parametrelerinin belirlenmesi için oluşturulan Tablo 3'den, 5 – 10 boyutunda olan problemlerde *tabu_süre_alt ve tabu_süre_üst* parametrelerinin $n/2$ ve $2n$ veya n ve $2n$ seçilebileceği anlaşılmaktadır. Bu çalışmada $n/2$ ve $2n$ değerleri tercih edilmiştir. Diğer taraftan, 20 boyutunda olan problemlerde en iyi başarı *tabu_süre_alt ve tabu_süre_üst* parametreleri $n/2$ ve $2n$ seçildiğinde sağlanırken, 30 boyutunda olan problemlerde $n/2$ ve n seçildiğinde en iyi başarıya ulaşılmaktadır.

Parametre değerleri belirlendikten sonra algoritmalar, 40 farklı KAP'nin her biri için 5'er kez denenmiştir. BTA ve KTA algoritmaları deterministik yapıya sahip olmalarına rağmen, başlangıç çözümlerinin rassal olarak seçilmesiyle denemeler gerçekleştirilmiştir. 5, 6, 7, 8 ve 9 tesisli problemler için BTA ve KTA algoritmalarının durdurulma koşulu olarak eniyi çözümün bulunması veya enbüyük tekrar sayısının 1.000 olması dikkate alınmıştır. Enbüyük tekrar sayısı, 10 boyutunda 1.500, 20 boyutunda 2.500 ve 30 boyutunda 3.500 olarak seçilmiştir. RA algoritmasının durdurulma koşulu olarak ise, örnek genişliğinin BTA ve KTA algoritmalarının ilgili problemlerde aradıkları ortalama çözüm sayısına eşit olması dikkate alınmıştır.

Tablo 4'de algoritmaların eniyi sonucu bilinen her problem için elde ettikleri ortalama sonuçlar verilirken, Tablo 5'de, Tablo 4'deki veriler problem boyutlarına göre ortalamalar alınarak biraz daha genelleştirilmiştir. Tablo 6'da yine küçük boyutlu problemler için eniyi yerleşim planları verilmektedir. Tablo 7'de ise, eniyi çözümü bilinmeyen büyük boyutlu problemler için ortalama sonuçlar görülmektedir. Bu tablolarda kullanılan kısaltmaların açık ifadesi şöyledir:

ÇUB : Çözüm Uzayının Büyüklüğü.
EMTM : Eniyi Malzeme Taşıma Maliyeti.
EMSY : Eniyi Maliyetten Sapma Yüzdesi
ÇUAY : Çözüm Uzayının Aranan Yüzdesi
BMTM : Bulunan Malzeme Taşıma Maliyeti

Tablo 4 incelendiğinde, BTA algoritmasının küçük boyutlu problemler için yapılan denemelerin hepsinde çözüm uzayının oldukça küçük bir kısmını arayarak eniyi çözümü bulabildiği görülmektedir. KTA algoritması ise 7/5 ve 8/4 numaralı problemler haricinde yine yapılan denemelerin hepsinde eniyi çözüme ulaşabilmektedir. Ancak KTA algoritmasının çözüm uzayını arama yüzdesi, problem boyutu büyüdükçe BTA algoritmasından daha yüksek bir değere ulaşmaktadır. Diğer taraftan RA algoritması, BTA ve KTA algoritmalarından hem eniyi çözümden sapma hem de çözüm uzayını arama yüzdesine göre çok daha kötü bir performans göstermektedir. 5 boyutunda olan problemler için RA algoritması tüm denemelerde eniyi çözümü bulabilmiş olmasına rağmen çözüm uzayının yaklaşık olarak tamamını aramıştır. Tablo 4'deki verilerin boyut bazında ortalamalarına göre oluşturulan Tablo 5, BTA algoritmasının tüm küçük problem boyutlarında KTA ve RA algoritmalarından daha yüksek bir başarıya sahip olduğunu göstermektedir.

Büyük boyutlu problemler üzerinde yapılan karşılaştırmalarda, RA algoritmasının tüm problem boyutlarında BTA ve KTA algoritmalarından çok daha kötü bir başarı düzeyine sahip olduğu Tablo 7'de görülmektedir. 10 boyutundaki problemlerde 5 numaralı problem hariç BTA ve KTA algoritmalarının başarı düzeyleri eşit olmaktadır. 5 numaralı problem için ise BTA algoritması, KTA algoritmasından daha etkindir. 20 boyutunda olan problemlerin hepsinde bulanık tabu sürelerinin kullanılmasının daha başarılı sonuçlar verdiği yine Tablo 7'de

Tablo 4. Küçük Boyutlu Tüm Test Problemleri Üzerine Algoritmaların Ortalama Başarıları.

Kareli Atama Problemi			BTA		KTA		RA	
Boyut/No	ÇUB	EMTM	EMSY%	ÇUAY%	EMSY%	ÇUAY%	EMSY%	ÇUAY%
5/1	120	317	0	11,83	0	9,5	0	100
5/2	120	5708	0	2,33	0	2,33	0	95,50
5/3	120	15557	0	3,83	0	3,83	0	100
5/4	120	9828	0	5,67	0	5,67	0	100
5/5	120	12198	0	4,67	0	4,67	0	89,01
6/1	720	18444	0	2,06	0	1,42	0,62	100
6/2	720	15978	0	1,61	0	1,50	0,27	88,64
6/3	720	19292	0	1,56	0	1,56	0,03	70,40
6/4	720	7953	0	3,64	0	6,19	1,36	94,58
6/5	720	17930	0	2,36	0	3,14	0,16	100
7/1	5040	14597	0	0,33	0	0,87	1,03	15,17
7/2	5040	26875	0	0,42	0	0,79	0,34	18,12
7/3	5040	24613	0	0,52	0	0,44	1,37	16,62
7/4	5040	24896	0	0,47	0	0,54	1,20	15,30
7/5	5040	21652	0	0,54	0,31	4,59	2,25	18,21
8/1	40320	34684	0	0,09	0	0,19	2,39	2,48
8/2	40320	32000	0	0,19	0	0,42	3,59	2,48
8/3	40320	28518	0	0,40	0	0,14	4,93	2,48
8/4	40320	31870	0	0,09	0,09	0,59	3,44	2,48
8/5	40320	25698	0	0,07	0	0,16	8,06	2,48
9/1	362880	30076	0	0,04	0	0,05	6,08	0,28
9/2	362880	42787	0	0,02	0	0,02	4,22	0,28
9/3	362880	34875	0	0,005	0	0,009	6,19	0,28
9/4	362880	36620	0	0,007	0	0,010	4,27	0,28
9/5	362880	42695	0	0,008	0	0,010	3,65	0,28

Tablo 5. Küçük Problem Boyutlarına Göre Algoritmaların Ortalama Başarıları.

	BTA		KTA		RA	
Boyut	Ort. EMSY	Ort. ÇUAY	Ort. EMSY	Ort. ÇUAY	Ort. EMSY	Ort. ÇUAY
5	0	5,667	0	5,200	0	96,902
6	0	2,246	0	2,762	0,488	90,724
7	0	0,456	0,062	1,446	1,238	16,684
8	0	0,168	0,018	0,300	4,482	2,480
9	0	0,016	0	0,020	4,882	0,280
Ortalama	0	1,711	0,016	1,946	2,218	41,414

Tablo 6. Küçük Boyutlu Problemler İçin Eniyi Yerleşim Planları.

Boyut/No	Eniyi yerleşim planı	Boyut/No	Eniyi yerleşim planı
5/1	5 1 3 2 4	7/4	2 1 5 7 6 3 4
5/2	1 5 2 4 3	7/5	7 6 2 4 1 3 5
5/3	4 3 2 5 1	8/1	2 5 7 4 3 1 8 6
5/4	3 5 1 4 2	8/2	7 8 3 6 5 1 4 2
5/5	3 1 4 5 2	8/3	5 1 8 6 4 3 7 2
6/1	3 4 1 2 5 6	8/4	7 6 5 1 2 4 3 8
6/2	2 1 3 4 6 5	8/5	6 5 2 1 4 7 3 8
6/3	4 1 3 2 5 6	9/1	1 3 6 9 4 8 5 7 2
6/4	3 4 2 1 5 6	9/2	5 3 9 8 7 1 4 6 2
6/5	6 1 2 3 4 5	9/3	9 1 4 7 5 3 6 2 8
7/1	7 6 5 4 2 1 3	9/4	8 2 1 6 4 7 3 9 5
7/2	1 7 4 5 3 6 2		
7/3	5 3 1 4 2 6 7	9/5	7 4 2 5 3 8 6 1 9

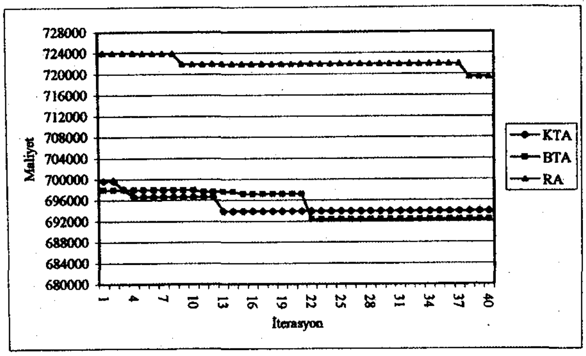
Tablo 7. Büyük Boyutlu Test Problemleri Üzerinde Algoritmaların Buldukları Ortalama Malzeme Taşıma Maliyeti.

Boyut/No	BTA	KTA	RA
10/1	45157,0	45157,0	47621,8
10/2	43470,0	43470,0	46094,4
10/3	44282,0	44282,0	47735,8
10/4	42252,0	42252,0	45416,0
10/5	54287,0	54347,8	57124,0
20/1	692881,6	692972,0	1050269,6
20/2	210781,0	211294,8	226100,4
20/3	211765,6	212208,4	228372,0
20/4	202466,2	202912,0	219157,8
20/5	204522,0	205455,0	221068,2
30/1	470554,4	471890,0	507328,4
30/2	482664,6	482770,6	519662,4
30/3	442195,6	443678,8	481238,0
30/4	356066,2	356505,6	390911,2
30/5	479557,2	479428,6	519897,0

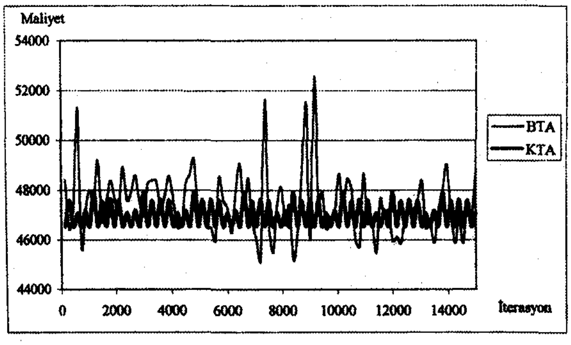
görülmektedir. Benzer şekilde, 30 boyutunda olan problemlerde 5 numaralı problem hariç, BTA algoritması KTA algoritmasından daha iyi çözümler bulabilmiştir. 30 boyutundaki 5 numaralı problemde ise KTA algoritmasının bulduğu çözümlerin ortalama maliyeti daha düşük olmuştur. Büyük boyutlu problemler için, RA algoritmasının oldukça kötü bir başarıya sahip olduğu, diğer taraftan BTA algoritmasının genel olarak KTA algoritmasında daha iyi çözümlere ulaşabildiği söylenebilir.

BTA, KTA ve RA algoritmalarının 20 tesisli bir yerleşim problemi üzerindeki yakınsama hızları Şekil 2'de görülmektedir. Görüldüğü gibi RA algoritması, BTA ve KTA algoritmalarından oldukça kötü bir çözüme yakınsamaktadır. BTA algoritması ise aramanın başlangıçlarında KTA algoritmasından daha yüksek maliyetli çözümler bulsa da, sonuçta KTA algoritmasının daha iyi çözümlere ulaşabilmektedir.

Şekil 3'de ise BTA ve KTA algoritmalarının 10 tesisli bir yerleşim problemi üzerindeki arama davranışları görülmektedir. Şekilden de anlaşıldığı gibi, KTA algoritmasında arama belli bölgelerde yoğunlaşmakta, BTA algoritmasında ise farklı bölgelere yönlenerek nihayetinde KTA algoritmasından daha iyi çözümlere ulaşılabilmektedir. Böylece, her iki algoritma arasındaki tek değişiklik bulanık tabu süresinin kullanımı olduğundan dolayı bulanık tabu sürelerinin etkinliği gösterilmiş olmaktadır.



Şekil 2. 20 Tesisli Bir Problem için BTA, KTA, RA Algoritmalarının Yakınsamaları.



Şekil 3. BTA ve KTA Algoritmalarının Arama Davranışları.

6. SONUÇ VE DEĞERLENDİRME

Tesis yerleşim probleminin kombinatoryal yapısı ve NP-tam olması, araştırmacıları tesis yerleşimi problemine sezgisel yaklaşımlar uygulamaya yönlendirmiştir. Genellikle kareli atama problemi olarak modellenen tesis yerleşimi problemine ilk uygulanan sezgisel yaklaşımlar kurucu ya da iyileştirici yapılara sahiptirler. Bu sezgiseller ya da bunların kombinasyonları, problem için ancak yerel bir eniyi çözüm bulabilmektedirler. Bu nedenle, genetik algoritmalar, tavlama benzetimi, tabu arama ve yapay sinir ağları gibi zeki sezgisel teknikler kareli atama problemine uygulanmış ve oldukça iyi sonuçlar elde edilmiştir.

Bu çalışmada ise bulanık tabu sürelerini kullanan bir tabu arama algoritması geliştirilmiş ve çeşitli boyutlardaki kareli atama problemleri üzerinde denenmiştir. Geliştirilen algoritma, bulanık tabu sürelerini kullanması açısından kaynakta görülen tabu arama yaklaşımlarından farklıdır. Geliştirilen tabu arama algoritması, sabit tabu süresini kullanan klasik tabu arama algoritmasıyla ve rassal arama algoritmasıyla karşılaştırılmıştır. Böylece bulanık tabu süresini kullanmanın daha iyi sonuçlar verip vermeyeceğinin gösterilmesi hedeflenmiştir. Elde edilen sonuçlar, bulanık-tabu arama algoritmasının hem bulduğu çözümlerin kalitesine göre hem de çözüm uzayında aradığı nokta sayısına göre diğer algoritmalardan daha üstün olduğunu göstermektedir.

Bu çalışmanın devamında, bulanık tabu sürelerini kullanan tabu arama algoritmasının diğer bazı NP-zor problemler üzerinde de etkin olup olmadığı incelenmesi amaçlanmaktadır. Ayrıca, tabu süreleri için farklı üyelik fonksiyonlarının algoritmanın başarısını artırıp artırmayacağı araştırılacaktır.

Ek 1. Bazı Küçük Boyutlu Problemler İçin Malzeme Taşıma Maliyet Matrisleri.

Problem 5/1 uzaklık ve akış matrisi:

	1	2	3	4	5
1	-	2	6	10	7
2		-	8	10	9
3			-	3	4
4				-	12
5					-

	1	2	3	4	5
1	-	0	3	5	10
2		-	11	0	6
3			-	8	9
4				-	7
5					-

Problem 6/2 uzaklık ve akış matrisi:

	1	2	3	4	5	6
1	-	97	22	92	11	57
2		-	56	12	70	60
3			-	52	13	15
4				-	23	24
5					-	76
6						-

	1	2	3	4	5	6
1	-	16	43	42	9	37
2		-	33	11	25	45
3			-	16	37	35
4				-	35	36
5					-	16
6						-

Problem 7/3 uzaklık ve akış matrisi:

	1	2	3	4	5	6	7
1	-	70	85	74	22	15	18
2		-	51	55	76	86	36
3			-	70	39	40	84
4				-	45	77	11
5					-	84	54
6						-	57
7							-

	1	2	3	4	5	6	7
1	-	45	18	16	16	19	2
2		-	32	24	36	2	33
3			-	32	12	16	36
4				-	17	22	44
5					-	31	32
6						-	45
7							-

Problem 8/4 uzaklık ve akış matrisi:

	1	2	3	4	5	6	7	8
1	-	83	13	66	71	48	28	46
2		-	91	72	94	62	58	81
3			-	31	44	53	58	64
4				-	93	17	64	44
5					-	28	67	97
6						-	92	53
7							-	26
8								-

	1	2	3	4	5	6	7	8
1	-	9	6	44	34	21	18	40
2		-	12	30	23	25	2	10
3			-	10	17	16	32	34
4				-	28	13	18	43
5					-	12	40	28
6						-	17	29
7							-	23
8								-

Problem 9/5 uzaklık ve akış matrisi:

	1	2	3	4	5	6	7	8	9
1	-	87	22	24	89	25	61	17	81
2		-	21	91	21	62	75	69	71
3			-	73	73	59	83	61	52
4				-	69	53	22	21	29
5					-	69	87	68	98
6						-	59	21	29
7							-	25	23
8								-	90
9									-

	1	2	3	4	5	6	7	8	9
1	-	16	25	17	32	31	37	29	2
2		-	3	43	33	23	36	20	31
3			-	44	23	30	12	29	11
4				-	6	7	9	22	18
5					-	20	40	30	24
6						-	29	25	17
7							-	45	31
8								-	44
9									-

KAYNAKÇA

Abninnour-Helm, S. ve Hadley, S.W. (2000). Tabu search based heuristics for multi-floor facility layout. *International Journal of Production Research*, 38(2), 365-383.

Al-Hakim, L. (2000). On solving facility layout problems using genetic algorithms. *International Journal of Production Research*, 38(11), 2573-2582.

Alvarenga, A.G., Negreiros-Gomes, F. ve Mestria, M. (2000). Metaheuristic methods for a class of the facility layout problems. *Journal of Intelligent Manufacturing*, 11, 421-430.

Badiru, A.B. ve Arif, A. (1996). FLEXPERT: Facility layout expert system using fuzzy linguistic relationship codes. *IIE Transactions*, 28(4), 295-308.

Bazaraa, M.S. ve Elshafei, A.N. (1979). An exact branch and bound procedure for quadratic assignment problems. *Naval Research Logistic Quarterly*, 26, 109-121.

Bazaraa, M.S. ve Sherali, A.N. (1980). Bender's partitioning scheme applied to a new formulation of the quadratic assignment problem. *Naval Research Logistic Quarterly*, 27(1), 29-41.

Bazargan, M. ve Kaebernick, H. (1997). An approach to the machine layout problem in a cellular manufacturing environment. *Production Planning & Control*, 8(1), 41-55.

Burkard, R.E. ve Stratmann, K.H. (1978). Numerical investigation on quadratic assignment problems. *Naval Research Logistic Quarterly*, 25, 129-148.

Burkard, R.E. (1990). Locations with spatial interactions: the quadratic assignment problem. *Discrete Location Theory*, Eds: P.B. Mirchandani, R.L. Francis, Wiley, Berlin.

Cerny, V. (1985). Thermodynamical approach to traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, 45(1), 41-51.

Chiang, W.C. ve Chiang, C. (1998). Intelligent local search strategies for solving facility layout problems. *European Journal of Operational Research*, 106(2-3), 457-488.

Chiang, W.C. ve Kouvelis, P. (1996). An improved tabu search heuristic for solving facility layout design problems. *International Journal of Production Research*, 34(9), 2565-2585.

Chittratanawat, S. ve Noble, J.S. (1999). An integrated approach for facility layout, P/D location and material handling system design. *International Journal of Production Research*, 37(3), 683-706.

Chung, Y.K. (1999). A neuro-based expert system for facility layout construction. *Journal of Intelligent Manufacturing*, 10(5), 359-385.

Chwif, L., Barretto, M.R.P. ve Moscato, L.A. (1998). A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36(1-2), 125-132.

De Jong, K.A. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* PhD Thesis, University of Michigan.

Foulds, L.R. (1983). Techniques for facilities layout: deciding which pairs of activities should be adjacent. *Management Science*, 29(12), 1414-1426.

Gau, K.Y. ve Meller, R.D. (1999). An iterative facility layout algorithm. *International Journal of Production Research*, 37(16), 3739-3758.

Gilmore, P.C. (1963). Optimal and suboptimal algorithms for the quadratic assignment problem. *SIAM Journal*, 10(2), 205-313.

Glover, F. (1989). Tabu search – part I. *ORSA Journal on Computing*, 1(3), 190-206.

Glover, F. (1990). Tabu search – part II. *ORSA Journal on Computing*, 2(1), 4-32.

Glover, F. ve Greenberg, H.J. (1989). New approaches for heuristic search: a bilateral linage with artificial intelligence. *European Journal of Operational Research*, 39, 119-130.

Glover, F. ve Laguna, M. (1993). *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Syntefic Publications.

Glover, F. ve Laguna, M. (1997). *Tabu Search*, Kluwer Academic Publishers, Boston.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

Hamamoto, S., Yih, Y. ve Salvendy, G. (1999). Development and validation of genetic algorithm-based facility layout-a case study in the pharmaceutical industry. *International Journal of Production Research*, 37(4), 749-768.

Hansen, P. (1986). The steepest ascent mildest descent heuristic for combinatorial programming, *Congress on Numerical Methods in Combinatorial Optimization,* Capri, Italy.

Hassanein, A.A. ve Cherlopalle, V. (1999). Fuzzy sets theory and range estimating. *AACE International Transactions*, 4, 1-9.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Jajodia, S., Minis, I., Harhalakis, G. ve Proth, J.M. (1992). CLASS: Computerized layout solutions using simulated annealing. *International Journal of Production Research*, 30(1), 95-108.

Kazurhiro, T., Sunil, B. ve Yoshiyasu, T. (1996). A neural network approach to facility layout problems. *European Journal of Operational Research*, 83(3), 556-570.

Kim, J.G. ve Kim Y.D. (1998). A space partitioning method for facility layout problems with shape constraints. *IIE Transactions*, 30(10), 947-957.

Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

Kochhar, J.S. ve Heragu, S.S. (1999). Facility layout design in a changing environment. *International Journal of Production Research*, 37(11), 2429-2446.

Koopmans, T.C. ve Beckman, M.J. (1957). Assignment problems and the location of economic activities. *Econometrica*, 25(1), 53-76.

Kusiak, A. (1990). *Intelligent Manufacturing Systems*. Prentice-Hall, Englewood Cliffs, NJ.

Lawler, E.L. (1963). The quadratic assignment problem. *Management Science*, 9(4), 586-599.

Liao, T.W. (1994). Design of line-type cellular manufacturing systems for minimum operating and material-handling costs. *International Journal of Production Researh*, 32(2), 387-397.

Mak, K.L., Wong, Y.S. ve Chan, F.T.S. (1998). A genetic algorithm for facility layout problems. *Computer Integrated Manufacturing Systems*, 11(1-2), 113-127.

McCulloch, W.S. ve Pitts, W. (1943). *A Logical Calculus of the Ideas Immanent in Nervous Activity*.

Meller, R.D. ve Bozer, Y.A. (1996). A new simulated annealing algorithm for the facility layout problem. *International Journal of Production Research*, 34(6), 1675-1692.

Raoot, A.D. ve Rakshit, A. (1994). A fuzzy heuristic for the quadratic assignment formulation to facility layout problem. *International Journal of Production Research*, 32(3), 563-581.

Resende, M.G.C., Ramakrishnan, K.G. ve Drezner, Z. (1995). Computing lower bound for the quadratic assignment problem with an interior point algorithm for linear programming. *Operations Research*, 43(5), 781-791.

Rosenblatt, F. (1962). *Principles of neurodynamics*. Washington, DC: Spartan Books.

Rossin, D.F., Springer, M.C. ve Klein, B.D. (1999). New complexity measures for the facility layout problem: an emprical study using traditional and neuras network analysis. *Computers & Industrial Engineering*, 36(3), 585-602.

Sule, D.R. (1988). *Manufacturing Facilities*. PWS-KENT, MA.

Suresh, G., Vinod, V.V. ve Sahu, S. (1995). A genetic algorithm for facility layout. *International Journal of Production Research*, 33(12), 3411-3423.

Tam, K.Y. ve Chan, S.K. (1998). Solving facility layout problems with geometric constraints using parellel genetic algorithms: experimentation and findings. *International Journal of Production Research*, 36(12), 3253-3272.

Wang, T.Y., Lin, H.C. ve Wu, K.B. (1998). An improved simulated annealing for facility layout problems in cellular manufacturing systems. *Computers & Industrial Engineering*, 34(2), 309-319.

Yip, P.P.C ve Pao, Y.H. (1994). A guided evolutionary simulated annealing approach to the quadratic assignment problem. *IEEE Transactions on System Man and Cybernetics*, 24(9), 1383-1387.

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-353.

Zhang, G.Q., Xue, J. ve Lai, K.K. (2000). A genetic algorithm based heuristic for adjacent paper-real layout problem. *International Journal of Production Research*, 38(14), 3343-3356.

Orhan Türkbey, 1949 yılında Ankara'da doğdu. İlk ve Orta öğrenimini Ankara'da tamamladıktan sonra sırasıyla; Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Makina Mühendislik Bölümü'nden Lisans derecesi (B.Sc), Ankara Üniversitesi Siyasal Bilimler Fakültesi İşletme Bölümü'nden *Yüksek Lisans derecesi (M.B.A)*, Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi İşletme Bölümü'nden *Lisans ve Yüksek Lisans derecesi (B.Sc) (M.B.A)*, Gazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalından "Endüstri İşletmelerinde İşyeri Yerleşim Düzenlemesi ve Bilgisayar Destekli Sezgisel Bir Yöntem (CRAFT) Uygulaması" isimli çalışma ile *Doktora derecesi (Ph.D)* aldı. Askerliğini Türk Hava Kuvvetleri Bilgi-İşlem Merkezi'nde Program Subayı olarak tamamladı. Kamu Kuruluşlarının çeşitli kademelerinde *(DPT Müsteşarlığı, T.D.Ç.İ. Gn Md. Başmühendis ve Grup Bşk. Yrd.'sı, T.C.Başbakanlık Danışmanı ve Ekonomik İşler Başkanı)* üst düzey görevlerde bulundu. Halen, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Endüstri Mühendisliği Bölümü'nde Öğretim Üyesi olarak görev yapmaktadır. Aynı Bölümde "Tesis Yerleşimi ve Düzenlemesi, İleri Tesis Planlaması, Matematiksel Modeller, Yöneylem Araştırması" konularında dersler vermektedir. Yabancı dili ingilizce olup, evli ve iki çocuk babasıdır.

Çiğdem Alabaş, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Endüstri Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır. Gazi Üniversitesi Endüstri Mühendisliği Bölümünde 1995 yılında lisans programını, 1999 yılında yüksek lisans programını tamamlamıştır. Halen aynı bölümde Araştırma Görevlisi olarak çalışmaktadır. Çiğdem ALABAŞ'ın çalışma konuları benzetim tekniği ile karmaşık sistemlerin modellenmesi ve modern sezgisel tekniklerdir.