Gazi Üniv. Müh. Mim. Fak. Der.
Cilt 18, No 4, 45-56, 2003

J. Fac. Eng. Arch. Gazi Univ.
Vol 18, No 4, 45-56, 2003

TESİS YERLEŞİM PROBLEMLERİNE SEZGİSEL METOTLARLA YAKLAŞIM

Vecihi YİĞİT ve Orhan TÜRKBEY
Endüstri Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Gazi Üniversitesi, Maltepe, 06570 Ankara, vyigit@gazi.edu.tr, turkbey@gazi.edu.tr

ÖZET

Yıllardan beri çalışılmakta olan Tesis Yerleşim Problemleri (TYP) imalat ve yöneylem araştırmasında zengin bir literatüre sahiptir ve önemli bir rol oynamaktadır. Temel Fabrika Yerleşim Problemi olarak da bilinen Sınırsız Kapasiteli Tesis Yerleşim Problemleri (SKTYP), TYP ailesinin bir üyesidir. SKTYP, NP-Hard yapıya sahip olduğundan dolayı, problem boyutu ne kadar artarsa optimal sonucu bulmakta o kadar zordur ve ayrıca güvenilir sonuçlara ulaşmak da bir hayli zaman gerektirir. Bu makale, Yöneylem Araştırması Kütüphanesinde bulunan ve karşılaştırma için çok iyi bilinen SKTYP'ni Tepe-Tırmanma ve Tavlama Benzetimi (TB) metotlarıyla incelemeyi hedeflemektedir. Elde edilen sonuçlar geliştirilen Tavlama Benzetimi (TB) Algoritmasının iyi sonuçlar verdiğini göstermektedir.

Anahtar Kelimeler: Sınırsız kapasiteli tesis yerleşim problemleri, optimizasyon, tepe-tırmanma, tavlama benzetimi

AN APPROACH TO THE FACILITY LOCATION PROBLEMS WITH HILL-CLIMBING AND SIMULATED ANNEALING

ABSTRACT

Facility location problems (FLP) have been studied for many years and there are lots of reported Works in the literature. FLP's play an important role in operation research and in manufacturing. The uncapacitated facility location (UFL) problems, also known as simple plant location problem, are basically a member of the family of facility location problems. Since FLP'S have NP-Hard nature, the larger the size of the problem, the harder to find the optimal solution and furthermore, the longer to reach a reasonable results. This paper focuses to the examination of hill-climbing and simulated annealing approaches for Uncapacitated Facility Location (UFL) problems with the problems tackled are very well known benchmarks that are accessible on the Operation Research Library. The results show that developed Simulated Annealing Algorithm gives good solution.

Keywords: Uncapacitated facility location problems, optimization, hill-climbing, simulated annealing

1. GİRİŞ

Günümüzde dünyanın küçülmesi, tüketim bilincinin artması ve kaynakların azalması gerek kamu gerek özel sektördeki işletmelerin ayakta kalabilme ve gelişebilmeleri için hem kendilerini yenilemeleri hem de rekabet ortamına ayak uydurmaları gerektiği gerçeğini ön plana çıkarmıştır. TYP, üretim ve dağıtım ağının tasarımı, dağıtılmış veri ve iletişim tasarımı, banka, sağlık, itfaiye gibi servis operasyonlarının tasarımı, elektrik iletim hatlarının tasarımı ve e-ticarette siparişleri karşılamada servis tasarımında önemli rol arzetmektedir [1]. Tesis ağının stratejik seçiminin de müşterilerin isteğini karşılamada, üretim ve işletme maliyetlerinin düşürülmesinde ve işletmelerin sistemlerinin gelişmesinde anlamlı bir etkiye sahip olduğu da gerçektir.

TYP, karesel optimizasyon problemleri içerisinde önemli bir yere sahip olup, çözüm zamanının, problem boyutuna bağlı olarak üstel artış gösterdiği çözülmesi zor NP-Hard problemlerdendir [2]. Bu tür problemler, problem boyutunun polinom fonksiyonu ile sınırlı olan hesaplama zamanı içinde çözülemezler. Bu nedenle bu tür problemlerin çözümünde, optimumu bulmayı garanti etmeyen ancak, çözüm zamanı polinom sınırlar içinde kalan sezgisel metotlardan yararlanılır. Sezgisellerin çoğu probleme özgü özelliğe sahiptirler ve bir problem için kullanılan bir sezgisel, başka bir problem için kullanılamamaktadır [3]. Bununla beraber, problemlere uygulanmaları açısından esnek ve daha genel özelliğe sahip olan tekniklere (Tavlama Benzetimi, Tabu Arama, Yapay Sinir Ağları, Genetik Algoritmalar) ilgi son yıllarda giderek artmıştır. Bu teknikler çok sayıda problemlere uygulanmışlar ve oldukça güçlü bulunmuşlardır. Bu makalede, TYP'nden Sınırsız Kapasiteli Tesis Yerleşim Problemlerinin çözümü Tepe-Tırmanma (T-T) metoduyla geliştirilen basit bir algoritma ve daha sonra Tavlama Benzetimi Algoritması ile incelenecektir.

2. PROBLEMİN TANIMI

Büyük pratik öneme sahip olan yerleşim karar problemleri, fabrikalar, depolar veya enerji iletim hatları gibi tesislerin yerleştirilmesidir. Yerleşim karar problemleri literatürde çalışılmış ve **fabrika yerleşim problemi** veya **tesis yerleşim problemi** olarak adlandırılmıştır [2]. TYP; *p-medyan, p-center, sınırsız kapasiteli tesis yerleşim ve karesel atama problemleri* olmak üzere genel olarak dört sınıfa ayrılırlar. SKTYP'de amaç, ürünlere olan talebi karşılayacak şekilde maliyeti asgari düzeye indirmek veya karı çoklamaktır. Genellikle tesisleri yerleştirmek için sabit maliyetler ve ürünlerin tesislerle müşteriler arasında dağıtımı için taşıma maliyetleri vardır. Problem, aday tesis belirli bir kapasiteye sahip olduğunda, sağlayabileceği en çok talebe göre şekilleneceğinden dolayı kapasite kısıtlı tesis yerleşimi problemi halini alır. Aday tesis için kapasite kısıtı olmadığında ise temel veya sınırsız kapasiteli tesis yerleşimi problemi halini alır [2,4].

Bu problemlerin matematiksel formülasyonu çözüm metodunun çıkarımında verimli olan karışık tamsayılı programlamadır. SKTYP'ni problemini formüle etmek istersek; tek ürün için verilen taleple I={1,....,m} müşteri seti ve J={1,....,n} tesislerin yerleştirileceği alanların kümesi olsun. Her bir $j \in J$ tesisinin sabit maliyeti f_j , tüm $i \in I$ müşterilerinin talebi b_i, ve j tesisinden i müşterisinin talebini karşılayan bilinen taşıma maliyeti de c_{ij}'dir. Ayrıca, mevcut tüm aday tesislerin kapasiteleri tüm müşterilerin taleplerini karşılamak için yeterlidir ve herbir müşterinin talebi tek tesisten karşılanacaktır. Böylece, problemi aşağıdaki şekilde formüle edilebilir:

$$Z = \min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} + \sum_{j=1}^{m} f_j y_j \quad (1)$$

$$\sum_{j=1}^{m} x_{ij} = 1, \text{ for } \forall\ i \in I; \quad (2)$$

$$0 \le x_{ij} \le y_j \text{ and } y_j \in \{0; 1\}; \qquad \text{for } \forall\ i \in I \text{ and } \forall\ j \in J; \quad (3)$$

Burada; X_{ij}, i müşterisinin talebini yalnızca j tesisinden karşılayabileceği, y_j ise j tesisinin kurulup kurulmayacağını (y_j=1 veya y_j=0) gösterir. (2) nolu denklemdeki kısıt, tüm müşterilerin taleplerinin karşılanacağını, (3) nolu denklemdeki kısıt ise müşterilerin sadece açık tesislerden yararlanabileceğini gösterir. Mevcut tüm aday tesislerin kapasitelerinin tüm müşterilerin taleplerini karşılamak için yeterli olduğu varsayıldığından dolayı müşterilerin talep hacimleri kural kaybı olmaksızın gözardı edilmiş (b_i=1) ve 2 nolu kısıt talep değişkeni göz önüne alınmaksızın oluşturulmuştur.

Bu modelle amaçlanan, "aday tesislerden kaç tanesi açık olsun / yerleştirilsin ki tüm müşterilerin talepleri minimum maliyetle karşılansın yani toplam maliyet (sabit ve değişken) en az olsun" düşüncesidir. Yukarıdaki matematiksel model UFL Problemlerinin çözümünün yanı sıra bazı problemlere de uygulanabilmektedir [2]. Bunlar da şöyle sıralanabilir:

- Banka hesapları yerleşim problemi
- Kümeleme analizleri
- Kilit-Kutu yerleşimi
- Ekonomik parti hacmi
- Makine çizelgeleme ve bilgi düzeltme
- Portföy yönetimi
- İletişim ağlarının tasarımı

Bu çalışmalara ilaveten son zamanlarda ağ tasarım problemleri ve araç rotalama problemlerinde de bu modelden yararlanılmaktadır.

3. LİTERATÜRÜN İNCELENMESİ

Sezgisel olarak yapılmış olan birinci çalışma Kuehn ve Hamburger'in yapmış olduğu yerleşim problemlerinin büyük bir kısmını tasarlamaya yönelik sezgisel yaklaşımdır [5]. Bu yaklaşım iki adımdan ibarettir. Birinci adımda, amaç fonksiyonunu artırarak en çoklayacak şekilde tesisleri sırasıyla açar, ve amaç fonksiyonu değerini düşüren yeni bir tesis eklendiğinde durur. Kuehn ve Hamburger bu adımı "adım ekleme" olarak isimlendirmişlerdir. Literatürde ise, bu yöntem her bir adımında azami getiri hedeflediğinden dolayı "açgözlü sezgisel" olarak adlandırılmaktadır. İkinci adımında ise "yok et ve adım değiştir" kuralı vardır. Bunun sebebi ise, ekonomik olmayan aday tesisi yokedip açgözlü sezgiselle altsırada seçilmiş olan tesislerden birini değiştirmektir. Sonra da bu olurlu çözümden başlayıp açık tesisle kapalı tesisi değiş-tokuş yaparak çözüme gider. Bu sezgisel algoritmada "içdeğişim sezgisel" olarak adlandırılmaktadır.

Açgözlü ve içdeğişim sezgiselleri birçok yaklaşım algoritmasının temelidir. Her ikisi de gerekli başlangıç olurlu çözümü veya olurlu çözümleri kullanarak kesin algoritmalara yardımcı olurlar [6]. Bununla birlikte kullanıcının sezgisel çözüm değerinin optimal çözüm değerinden uzaklaşmadığı noktasında tatmin olması için, nihai çözümde sezgisellerin kullanımında üst sınır olmalıdır [7]. Cornuejols tarafından açgözlü ve içdeğişim sezgiselleri için bazı sınır değerler verilmiştir [8].

Tamsayılı programla ile optimum çözümü bulmadaki genel çözüm teknikleri SKTYP için özelleştirilmiştir. Karışık tamsayılı doğrusal programlama formülasyonunun çözümü Balinski ve Wolfe tarafından hazırlanan Bender eşitsizlikleri yaklaşımı ile incelenmiştir. Bu yaklaşım, SKTY problemini çözmek için ilk yaklaşımdır [6]. Magnanti ve Wong, Bender eşitsizliklerinin birleşimini hızlandırmak için teknik geliştirmişlerdir. Geliştirilen metotla Bender kesmelerinin olurlu kümesinden güçlü kesmeler üreterek tamsayılı programlama kısıt sayısı düşürülmüştür [2,4,9]. Nemhauser ve Wolsey alt modüler küme fonksiyonunun maksimizasyonu çalışmalarında Bender'in kesmelerini kullanmışlardır [2,4]. Daha sonra Guignard Bender eşitsizliklerinden üretilen Lagrangean ikili tırmanma yöntemini kullanarak SKTYP'nin ayrılabilir Lagrangean gevşetmesini kuvvetlendirmiş ve buna dayalı bir çözüm yöntemi hazırlamıştır [6].

Christofides ve Beasley, SKTYP'i için Lagrangean gevşetmeyi kullanmış ve (50*150) boyutundaki problem üzerinde sonuçlarını göstermiştir [11]. Beasley daha sonra geliştirdiği algoritmayı literatürde yer alan problemlerin [12] temelini teşkil eden problemler üzerinde göstermiştir [13,14].

TYP'ne diğer bir yaklaşım düşüncesi ise, karışık tamsayılı programlama modelinin ikilisinin çözümünün araştırılması düşüncesidir [10,15,16]. SKTYP'nin ikilisini çözecek algoritmalar üst sınırlar için önemli avantaja sahiptir ve herhangi ikili olurlu çözümden çıkarılabilir [10,15]. Erlenkotter bu sınırı sağlamak için doğrusal programlamanın gevşekliği tamamlayıcı özelliğini kullanarak adım adım ileriye gitmiştir. Bu algoritma basit tırmanma ve düzeltme yöntemini değerlendiren doğrusal programlamanın ikili formülasyonuna dayanmaktadır [15]. Simao ve Thizy ise TYP'nin kapsayıcı formülasyonu temelinde ikili simpleks algoritmayı kullanılabilir hale getirmeyi tasarlamışlar ve standart veri kümeleriyle yapmış oldukları çalışmanın ikili yaklaşımların en iyisi olduğunu belirtmişlerdir [17]. Körkel ise Erlenkotter'in geliştirdiği algoritmanın ilkil-ikili gösteriminin nasıl iyileştirilebileceğini göstermiştir [10]. Yaptığı çalışmalar, büyük boyutlu problemler için hızlı ve anlamlıdır. Fakat bu ticari ürün pratik olarak kullanışlı değildir [18]. Coon ve Cornuejols ise ortogonal yansımalar yoluyla Doğrusal Programlamanın kısaltılmış ikilisinin tam çözüm olduğu fikrine dayalı bir çalışma yapmışlardır [9].

Modern Sezgisel Metotların uygulandığı ilk çalışma ise, Vaithyanathan ve arkadaşları'nın yaptığı çalışmadır. Bu çalışmada yapay sinir ağlarına tabu arama mekanizmasının kısa dönem hafızası dahil edilerek Gezgin Satıcı ve SKTYP'nin çözümü irdelenmiştir. Geliştirilen metotla tabu aramanın temel prensipleri yapay sinir ağlarına dahil edilmiş ve bu yönteme de Tabu Aramalı Yapay Sinir Ağı (TANN) denilmiştir [19]. Test eedilen problem boyutları 10*10, 20*20 ve 30*15 büyüklüğündedir ve optimal sonuçtan 20% sapma göstermektedir ki bu da pratik olarak anlamsızdır.

Kratica ve arkadaşları [18] tarafından ise genetik algoritma ile SKTYP'ni çözme yoluna gidilmiş ve geliştirilen algoritma ile 1000*1000 boyutunda müşteri ve tesis için problem çözümünü gerçekleştirilmiştir. Yapılan çalışmada literatürdeki problemleri incelenmiş ve sonuçlar Erlenkotter'in ikili tabanlı algoritması ile karşılaştırılmıştır.

4. SEZGİSEL METOTLAR

4.1. Tepe-Tırmanma (T-T) Metodu

T-T metodu bir iteratif iyileştirme (yerel arama) yöntemidir. T-T, iniş ya da adım adım iniş strateji olarak da adlandırılmaktadır. Bu metodun temelinde, tanımlanan bazı kurallara göre bir çözümden bir diğer komşu çözüme ulaşma vardır. T-T metodunun etkinliğinde iyi bir komşuluk yapısı seçiminin önemi büyüktür. İyi mutlak açıdan en iyi olmak zorunda değildir. T-T'nin zayıf yanı yerel ve genel en iyi arasındaki ayrımı yapamaması sonucu yerel optimumdan kaçamamasıdır. Yerel optimumdan genel optimuma geçebilmek için modern sezgiseller geliştirilmiştir. Kısaca T-T Algoritmasının adımları aşağıdaki gibi gösterilebilir:

1. Başlangıç çözümü üret; x_0: mevcut çözüm ve $x_0 \in R$

2. Aşağıdaki adımları tekrarla:
   a) $N(x_n)$ komşu seti içinden en iyi x' komşusunu seç.
   b) $f(x') \leq f(x_n)$ ise x' çözümünü yeni mevcut çözüm olarak ata: $x_{n+1} = x'$
   c) Aksi halde dur.

4.2. Tavlama Benzetimi Metodu

Tavlama Benzetimi (TB), karesel optimizasyon problemleri için iyi çözümler veren olasılıklı karar verme yöntemi olarak görülebilir. Kontrol parametresi "sıcaklık"tır ve minimizasyon problemleri için çözümün daha iyi bir çözüme ulaşmasının kabulu olasılığını değerlendirir. "Tavlama Benzetimi" ismi, katıların fiziksel tavlanma süreci ile olan benzerlikten ileri gelmektedir. TB algoritması, birbirlerinden bağımsız olarak, Kirkpatrick, Gelatt ve Vecchi (1983) ve Cerny (1985) tarafından ortaya konmuştur. Günümüze kadar, bilgisayar tasarımı, görüntü işleme, moleküler fizik ve kimya, çizelgeleme gibi farklı alanlardaki bir çok optimizasyon problemine uygulanmıştır [3].

Tüm çözümlerin sonlu kümesi S çözüm uzayı ve S'in üyeleri için tanımlanan maliyet değerleri f maliyet fonksiyonu ile tanımlanacak olursa, söz konusu problem, tüm S üzerinde f'_s'i enküçükleyen ve $s \in S$ olan bir s çözümünün veya durumunun bulunmasıdır. Varsayılsın ki, s_n ve s_n' n. iterasyonda bulunmuş çözümler ve (n+1). iterasyona girecek çözümü ($s_{n+}1$) belirlenmek istensin. s_n', s_n'in en iyi komşu çözümüdür ve komşuluk fonksiyonuyla elde edilir. Yeni en iyi çözüm aşağıdaki gibi ifade edilebilir:

$$s_{n+1} \leftarrow \begin{cases} s_n' & \Delta s < 0 \\ s_n' & e^{-\Delta s / t_n} > r_n \\ s_n & \text{aksi halde} \end{cases} \tag{4}$$

Burada, $\Delta s = s_n' - s_n$, r_n yeni çözüme stokastik karar vermek için üretilmiş rassal sayı ve t_n ise sıcaklık fonksiyonuyla ($T = f(t_n)$) soğutulan başlangıç sıcaklığının n. iterasyondaki sıcaklık seviyesidir. Yani, yeni çözümü (s_n') sonraki iterasyona taşımak istiyorsak, ya eski çözümden (s_n) daha iyi olması ya da daha kötü olmasına rağmen stokastik kuralı tatmin etmesi gerekmektedir. Stokastik kuralın tatmin edilmesinin sebebi, optimizasyon sürecinin yerel minimumdan kurtulmasını sağlamasıdır. Bu da Tavlama Benzetiminin ana özelliklerinden birisidir. Çünkü, Tavlama Benzetimi'nde hedef yerel optimumdan kurtulup genel optimuma ulaşmaktır. Komşu çözüme giderken Δs kadar bir yükselmeye yol açan hareketin t_n sıcaklığındaki kabul edilme olasılığı, kabul fonksiyonu $\left(e^{-\Delta s / tn} \right)$ olarak adlandırılır. Kabul fonksiyonuna göre, çözümde meydana gelen küçük artışların kabul edilme olasılığı, büyük artışların kabul edilme olasılığından daha fazladır. Ayrıca, T yüksek olduğunda hareketlerin çoğu kabul edilecektir. T sıfıra yaklaştığında ise, çözümde artışa yol açan hareketlerin çoğu reddedilecektir. Bu nedenle TB algoritmasında, yerel optimum tuzaklarına düşülmesini engellemek için göreceli olarak yüksek bir T değeri ile aramaya başlanır. TB algoritması, bir taraftan sıcaklık yavaş yavaş azaltılırken, her sıcaklık değerinde belli sayıda hareket deneyerek arama işlemini sürdürür. Geliştirilen Tavlama Benzetimi algoritmasının adımları Şekil 1'de görülmektedir. Algoritma önceden tanımlanmış iterasyon sayısına bağlıdır. İlk önce rassal olarak başlangıç çözüm kümeleri üretilmekte ve kaç iterasyonda sonlandırılacağı belirlenmektedir. Daha sonra TB en yüksek ısı (100) ile başlamakta ve *t=0.995t* fonksiyonu ile ısı 0,01'den küçük olana kadar soğutulmaktadır. Böylece, seçilen ve işlenen çözüm kümesi, tekrardan çözüm kümeleri içine diğerleriyle kıyaslanmak için konmaktadır. Daha sonra çözüm kümeleri içindeki diğer herhangi bir çözüm rassal olarak seçilmekte ve işlenmektedir.

Begin
- *Mümkün bir çözüm seç* (***eski***),
- *Başlangıç sıcaklığını seç (t=100),*
- *Sıcaklık azatlım fonksiyonunu seç f(t),*

Repeat:

repeat:-
- *Komşu arama fonksiyonuna göre en iyi komşuyu seç*
- *Yeni çözümü hesapla* (***yeni***)
- ***if*** (***yeni-eski***)<0 then *yeniyi eskiyle değiştir*
- ***else***
  - *düzgün dağılıma sahip rassal sayı üret* ***(r)***
  - ***if exp(-(yeni-eski)/t)>r*** then *yeniyi eskiyle değiştir*
  - ***endif***
- ***endif***

until *önceden belirlenmiş iterasyon sayısı kadar*
- *t=f(t)*

Until (*t<0.0) durdurma koşulu sağlanana kadar*

End

Şekil 1. Tavlama Benzetimi algoritması

4.2.1. Gösterim ve komşuluk fonksiyonu

SKTY Problemleri genellikle çift değişken yoluyla (tesis açıksa 1, kapalıysa 0) gösterilmiştir. Bir çözümden diğer çözüme hareket etmek için değişkenler rahatlıkla biri birine çevrilebilir. Sezgisel yöntemlerdeki kullanımlara bağlı olarak bu değişim bu çalışmada kullanılan komşuluk fonksiyonu ile veya genetik operatörlerle yapılabilir. Burada yerleştirilecek tesislerin kümesini tamsayı olarak ve (F) ile gösterilmiştir. Örneğin F={1,6,12,18}, 1,6,12 ve 18 nolu tesislerin açık olduğu

-yerleşeceği- ve diğerlerinin kapalı olduğu -yerleşmeyeceği- anlamına gelmektedir. Daha önce belirtildiği gibi, t zamanındaki çözüm durumunu s_t ile gösterilebilir ve herhangi bir çözüm durumu aşağıdaki gibi olabilir:

$S_t=\{F\}=\{f_t \mid 0 < i \leq m\}$, burada m maksimum tesis sayısıdır.

Bu çalışmada kullanılan komşuluk fonksiyonu ise, 3 farklı operatörle mevcut çözüm durumundan komşu çözümlere geçmeye izin vermektedir. Bu operatörler; mevcut çözüm kümesindeki tesislerden aşık olanı kapalı, kapalı olanı açık yapmak (exchange), çözüm kümesine yeni bir tesis eklemek (add) veya çözüm kümesinden bir tesis çıkarmak (remove) şeklindedir. Bu seçimlerle herhangi bir operatör rassal olarak aşağıdaki kurala göre seçilmektedir:

$$\text{operator} \longleftarrow \begin{cases} \text{Exchange()} & (\lambda(F)=1 \cap 0 \leq \rho < 0.7) \cup (\lambda(F) > 1 \cap 0 \leq \rho \leq 0.5) \\ \text{Add()} & (\lambda(F)=1 \cap 0.7 \leq \rho < 1) \cup (\lambda(F) > 1 \cap 0.5 \leq \rho \leq 0.7) \\ \text{Remove()} & (\lambda(F)=m) \cup (\lambda(F) < m \cap 0.7 \leq \rho \leq 1) \end{cases}$$

Burada $\lambda(F)$ çözüm kümesinin uzunluğu, ρ düzgün dağılımdan üretilmiş rassal sayı, ve m daha önce belirtildiği gibi en fazla tesis sayısıdır. Bu fonksiyonun özelliği ise, operatörün uygun ve hareket edilebilir olurlu çözüme gitmesini sağlayan koruyucu komşuluk fonksiyonu olmasıdır. Herhangi bir komşu çözümün uygunluğu çözüm kümesinin uzunluğu ve rassal sayı ile belirlenir ($\lambda(F)$, ρ). *Exchange()* and *Add()* $\lambda(F)=1$ olduğunda çalışır, dolayısıyla, *Remove()* sadece $\lambda(F)=m$ olduğunda çalışır. Bunun sebebi ise, hiçbir tesisin yerleştirilmemesi gibi olurlu olmayan bir çözüm kümesine gitmeyi engellemektir. Her üç operatörde rassal olarak aday tesisi çözüm kümesine alacaktır ve yeniden aynı aday çözüm kümesini üretmeyecektir. Komşuluk fonksiyonu seçilen ihtimallerle uygun operatörü seçiyor olmasına rağmen, çıktısı düzgün dağılım göstermektedir. Dolayısıyla, komşu çözüme hareket etmek düzgün dağılıma göre şekillenmektedir.

5. DENEYSEL ÇALIŞMALAR

Bu deneysel çalışma AMD (Athlon 1.7 GHz işlemci, 128 MB RAM) bilgisayarda, Windows XP altında yapılmıştır. Tüm yazılımlar Sun Java JDK1.3.1. dili kullanılarak kodlanmıştır. Örnek problemler ise literatürde yer alan [12] problemlerdir ve özellikleri Tablo 1'dedir. Tablo 1'de sırasıyla, problem boyutları (n tesis, m müşteri), optimum değerleri ve dosya büyülükleri vardır.

Tablo 2'de ise T-T metoduyla çözülen problemlerin sonuçları vardır. Her bir problem 40 defa çalıştırılmış ve çözüm sonuçlarının ortalamaları alınmıştır. Burada problemlerin hepsi çözülmemiştir. Çünkü T-T metodu optimum çözümü bulamamakta ve problem boyutu arttıkça sapmalarda artmaktadır. Bu çalışmada bu metodun incelenmesindeki amaç problemlerin keskin -zor aşılır- yerel optimumları

Tablo 1. Örnek problemler ve özellikleri

Problem Adı	**Problem Boyutu**	**Optimum Sonuç**	**Dosya Büyüklüğü**
71-72-73-74	16 X 50	932615.75, 977799.4 1010641.450 1034976.975	11 KB
101-102-103-104	25 X 50	796648.438 854704.200 893782.113 928941.750	16KB
131-132-133-134	50 X 50	793439.563 851495.325 893076.713 928941.750	32KB
A-B-C	100 X 1000	17156454.478 12979071.581 11505594.329	1,2 MB

Tablo 2. T-T Metodu ile bulunan çözüm sonuçları

Problem No	Bulunan Ortalama Sonuç	Fark	% Fark
71	942523.116	9907.366	0.011
72	999032.680	21233.280	0.022
73	1046462.978	35821.528	0.035
74	1109616.562	74639.587	0.072
101	826504.448	29856.010	0.037
102	900135.687	45431.487	0.053
103	958853.898	65071.785	0.073
104	1028611.260	99669.510	0.107
131	889106.148	95666.586	0.121
A	21010191.297	3853736.819	0.225

varsa test edilmesi ve geliştirilen TB Algoritmasında bu yerel optimumlardan kurtulmaya çalışmak içindir. Tablo 3'de ise TB ile bulunan sonuçlar vardır. İki farklı sonucun olmasının nedeni iterasyon sayısının her birinde farklı olmasıdır. Bulunan birinci sonuçlarda iterasyon sayısı her problem için problem boyutunun iki katıyken (örneğin problem A için 100*1000*2) diğerinde ise 4 katıdır. Her iki TB algoritmasında da 100 derece sıcaklıkla başlanmış ve $f(t)=0{,}995t$ ile $t<0{,}01$ oluncaya kadar soğutulmuştur. Tablo 1'de bulunan her bir problem her iki algoritma ile de rassal olarak 40'ar kez çözülmüştür ve sonuçlar 40 gözlemin ortalamasıdır.

Aşağıda bulunan Tablo 3'e bakıldığında, dikkat edilirse iterasyon sayısının artması çözüm kalitesini etkilememekte fakat çözüm zamanını düzgün olarak (yaklaşık iki katı) artırmaktadır. Büyük boyutlu problemlerin (A,B,C) çözümüne bakıldığında ise; A probleminin optimum sonucuna her iki algoritma ile de ulaşılmasına rağmen B ve C problemleri için mümkün olmamıştır. Sapma olan problemlerde kaç defa optimum sonucun bulunduğu ise Tablo 4'tedir.

Tablo 3. TB metodu ile bulunan çözüm sonuçları

Problem No	Bulunan Ortalama Sonuç		Fark		% Fark		Ortalama Zaman (ms)	
			SA1	SA2	SA1	SA2	SA1	SA2
71	932615.75	932615.75	0.000	0.000	0	0	131.0	212.05
72	977799.4	977799.4	0.000	0.000	0	0	69.14	137.00
73	1010903.67	1011165.89	262.218	524.436	0.00025	0.00051	44.00	80.38
74	1034976.98	1034976.98	0.000	0.000	0	0	31.90	66.95
101	797180.996	797058.098	532.559	409.661	0.00066	0.00051	160.7	332.57
102	854704.2	854704.2	0.000	0.000	0	0	125.76	258.24
103	894493.584	894445.058	711.471	662.945	0.00079	0.00074	79.86	165.24
104	928941.75	928941.75	0.000	0.000	0	0	55.71	115.43
131	793972.121	793808.257	532.559	368.695	0.00067	0.00046	325.90	662.10
132	851495.325	851495.325	0.000	0.000	0	0	250.05	509.71
133	893718.71	894013.586	641.998	936.874	0.00071	0.00104	186.71	372.00
134	928941.75	928941.75	0.000	0.000	0	0	117.52	218.71
A	17156454.5	17156454.5	0.000	0.000	0	0	79284.1	157422
B	13026014.5	13010843.6	46942.8	31771.9	0.00361	0.00244	126890	250460
C	11545398.5	11574901.7	39804.1	69307.3	0.00346	0.00602	163340	323291

Tablo 4. Sapma olan problemlerin analizi

Problem No	Çalıştırma Sayısı	Optimum		Sapma<0,01		Sapma<0,02		Sapma>0,02	
		SA1	SA2	SA1	SA2	SA1	SA2	SA1	SA2
73	40	35	34	5	6	-	-	-	-
101	40	16	22	24	18	24	-	-	-
103	40	7	8	33	32		-	-	-
131	40	17	16	24	24	16	-	-	-
133	40	19	5	21	35		-	-	-
B	40	22	29	16	10	2	1	-	-
C	40	4	2	30	30	2	6	4	2

6. SONUÇ

Literatüre bakıldığında, Tesis Yerleşim Problemlerinin imalat ve yöneylem araştırmasında zengin bir literatüre sahip olduğu ve önemli bir rol oynadığı görülür. Zaten Tesis Yerleşim Problemlerinin birçoğu NP-Hard problemlerdir ve bu problemleri çözmek için gerekli hesaplama zamanı eksponansiyel olarak artmaktadır. Sınırsız Kapasiteli Tesis Yerleşim Problemleri'nin çözümü için Tamsayılı Programlama yaklaşımı küçük boyutlu problemlerde iyi olmakla birlikte esnek olmaması ve problem açılımları karşısında yapısında tamamen köklü değişiklikler gerektirmesi nedeniyle avantajlı değildir. Problem boyutu arttıkça sezgisel yöntemler, Tamsayılı Programlamaya nazaran daha avantajlıdırlar. Sezgisel yöntemlerle büyük boyutlu problemlerin çözümüne gidilmiş ve değişik çalışmalar yapılmıştır. Bölüm 1'de belirtilen güçlü sezgisel algoritmalarla yapılan çalışmalara bakıldığında iki çalışma görülmektedir. Yapay sinir ağlarının tabu arama ile hibrit edildiği çalışmada, küçük boyutlu problemlerin çözümünde bile optimal sonuçtan uzaklaşıldığı ve iyi bir verim göstermediği görülür. Genetik algoritma yöntemiyle yapılan diğer çalışmada ise problem boyutu arttıkça optimumdan sapma oranının arttığı görülmüştür. Bu makalede basitçe TB ile SKTYP'nin çözümü irdelenmeye çalışılmıştır. Bundan sonra yapılabilecek çalışmalar, hem optimum sonuca ulaşmak hem de zaman açısından anlamlı olabilmelidir.

KAYNAKLAR

1. Francis, R.L., McGinnis L.F. and White, J.A., "Locational Analysis", **European Journal of Operations Research**, No 12**,** 220-252, 1983.
2. Mirchandani P.B. and Francis R.L., **Discrete Location Theory**, John Wiley & Sons, New York, A.B.D., 1990.
3. Reeves, Colin, **Modern Heuristic Techniques for Combinatorial Problems**, John Wiley & Sons, New York, A.B.D., 1993.
4. Krarup J., ve Pruzan, P.M., "The Simple Plant Location Problem: Survey and Synthesis", **European Journal of Operations Research,** No 12, 36-81, 1983.
5. Kuehn, A.A., and Hamburger, M.J., "A Heuristic Program for Locating Warehouses", **Management Sciences**, No 9, 643–666, 1963.
6. Guignard, M., "A Lagrangean Dual Ascent Algorithm for Simple Plant Location Problems", **European Journal of Operations Research, No** 35, 193-200, 1988.
7. Beasley, J.E., "Lagrangean Heuristic for Location Problems", **European Journal of Operations Research,** No 65, 383-399, 1993.
8. Cornuejols, G. and Thizy, J. M., "Some Facets of the Simple Plant Location Problem", **Mathematical Programming,** No 23, 50-74, 1982.
9. Conn A.R. and Cornuejols, G.A., "Projection Method for the Uncapacitated Facility Location Problem", **Mathematical Programming**, No 46, 273-298, 1990.
10. Koerkel, M., "On the Exact Solution of Large-Scale Simple Plant Location Problems", **European Journal of Operations Research**, No 39, 157-173, 1989.
11. Christofides, N., and Beasley J.E., "A Tree Search Algorithm for the P-Median Problem", **European Journal of Operations Research**, No 10, 196-204, 1982.
12. **http://mscmga.ms.ic.ac.uk/info.html**
13. Beasley, J.E., "An Algorithm for Solving Large Scale Capacitated Warehouse Location Problems", **European Journal of Operations Research**, No 33, 314-325, 1988.
14. Beasley, J.E., "Lagrangean Heuristics for Location Problems", **European Journal of Operations Research**, 65, 383-399, 1993.

15. Erlenkotter, D.A., "Dual-Based Procedure for Uncapacitated Facility Location", **Operations Researchs**, No 26**,** 992-1009, 1978.
16. Gao, L.L., Robinson E. and Powell, Jr., "Uncapacitated Facility Location: General Solution Procedure and Computational Experience", **European Journal of Operations Research**, No 76, 410-427, 1994.
17. Simao H.P. and Thizy, J.M., "A Dual Simplex Algorithm for the Canonical Representation of the Uncapacitated Facility Location Problem", **Operations Research Letters**, No 8, 279-286, 1989.
18. Kratica J., Tosic D., Filipovic V., Ljubic I., "Solving the Simple Plant Location Problem by Genetic Algorithms", **RAIRO - Operations Research**, Vol 35, No. 1, 127-142, 2001.
19. Vaithyanathan, S., Burke, L. and Magent, M.A., "Massively Parallel Analog Tabu Search Using Neural Networks Applied to Simple Plant Location Problem", **European Journal of Operations Research**, No 93, 317-330, 1996.