

İKİ AMAÇLI TESİS YERLEŞİMİ PROBLEMİ İÇİN BİR YİNELEMELİ YEREL ARAMA ALGORİTMASI

Mustafa AVCI^{1*}

¹İzmir Bakırçay Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü, İzmir
ORCID No: <http://orcid.org/0000-0001-9049-8292>

Anahtar Kelimeler	Öz
Yinelemeli Yerel Arama, Tesis Yerleşimi, Metasezgiseller	<i>Tesis yerleşimi, modern üretim sistemlerinde karşılaşılan en önemli sorunlardan biridir. Bu çalışmada hem nicel hem de nitel hedeflerin birleştirildiği iki amaçlı bir tesis yerleşim problemi (İA-TYP) ele alınmaktadır. Burada nicel amaç toplam malzeme taşıma maliyetinin en aza indirgenmesidir. Nitel amacımız ise toplam yakınlık derecelendirme puanlarının maksimize edilmesidir. Problemi çözmek için Değişken Komşuluklu İniş (DKİ) algoritmasının bir saptırma mekanizması ile birleştirildiği bir Yinelemeli Yerel Arama (YYA) algoritması önerilmiştir. Önerilen algoritmanın performansı önceki çalışmalarda sunulan çözüm algoritmalarının elde ettikleri çözümler baz alınarak değerlendirilmiştir. Hesaplama sonuçları, önerilen algoritmanın İA-TYP örneklerine yüksek kaliteli çözümler üretebildiğini göstermektedir.</i>

AN ITERATED LOCAL SEARCH ALGORITHM FOR THE BI-OBJECTIVE FACILITY LAYOUT PROBLEM

Keywords	Abstract
Iterated Local Search, Facility Layout, Metaheuristics	<i>Facility layout is one of the most important problems encountered in modern manufacturing systems. In this paper, we address a bi-objective facility layout problem (BO-FLP) in which both the quantitative and the qualitative objectives are combined. The quantitative objective is to minimize the total material handling cost while the qualitative one is the maximization of total closeness rating scores. In order to solve the problem, an Iterated Local Search (ILS) algorithm in which a Variable Neighborhood Descent (VND) algorithm is combined with a perturbation mechanism is proposed. The performance of the proposed algorithm is evaluated using the results published in the previous works. The computational results show that the proposed algorithm is able to produce high-quality solutions for the BO-FLP.</i>

Araştırma Makalesi	Research Article
Başvuru Tarihi : 18.02.2021	Submission Date : 18.02.2021
Kabul Tarihi : 17.06.2021	Accepted Date : 17.06.2021

*Sorumlu yazar; e-posta : mustafa.avci@bakircay.edu.tr

1. Giriş

Tesis yerleşimi, hem üretim hem de hizmet sistemlerinin etkili ve verimli bir şekilde çalışabilmesi için kritik bir öneme sahiptir. Bundan dolayı Tesis Yerleşimi Problemi (TYP) literatürde en çok çalışılan optimizasyon problemleri arasında yer almaktadır. TYP'yi çözmenin amacı bir üretim veya hizmet sistemi içerisinde yer alan fiziki kaynakların (bölüm, makine vb.) en yüksek faydayı sağlayacak şekilde konumlandırılmasını sağlamaktır. Tompkins, White, Bozer ve Tanchoco (2010), üretimdeki toplam işletme giderlerinin %20-50'sinin malzeme taşıma faaliyetlerinden kaynaklandığını belirtmektedir. Ayrıca, etkili bir tesis planlama sayesinde malzeme taşıma maliyetleri %10-30 arasında azaltılabilir (Tompkins ve diğ., 2010; Şahin, 2011).

TYP geleneksel olarak nicel ve nitel olmak üzere iki farklı amaç gözetilerek ele alınmaktadır. Birinci yaklaşımda amaç toplam malzeme taşıma maliyetinin en aza indirgenmesidir. İkinci yaklaşımda ise bir yakınlık fonksiyonu baz alınarak kaynaklar arasındaki toplam yakınlık derecelendirme puanının maksimize edilmesi amaçlanmaktadır. TYP hakkında detaylı literatür incelemeleri Singh ve Sharma (2006) ve Hosseini-Nasab, Fereidouni, Ghomi ve Fakhrzad (2018) tarafından yapılan çalışmalarda yer almaktadır.

TYP'nin gerçek hayat örnekleri incelendiğinde genellikle tek bir amaçtan ziyade hem nitel hem de nicel amaçların beraber gözetildiği görülmektedir. Bunun sebebi tek bir amaç dikkate alınarak oluşturulacak tesis yerleşimlerinin diğer amaç açısından tatmin edici bir sonuç vermeme ihtimalidir. Örneğin, yalnızca nicel amaç dikkate alınarak elde edilecek yerleşimle malzeme taşıma maliyetleri en aza indirgenebilir, fakat bu durumda bazı kaynakların birbirine yakın veya uzak konumlanması gibi birtakım gereksinimler karşılanamayabilir.

Bu çalışmada İki Amaçlı TYP (İA-TYP) için bir Yinelemeli Yerel Arama (YYA) algoritması önerilmiştir. YYA, başlangıç çözümü üretme, yerel arama ve saptırma olmak üzere 3 temel mekanizma içeren basit yapıda fakat etkili bir metasezgisel algoritmadır. YYA'nın temel çalışma prensibi şu şekilde işlemektedir: Yerel arama bir yerel optimum noktaya takıldığında mevcut çözüm üzerine bir saptırma uygulanarak yeni bir çözüme ulaşılmaktadır ve yerel arama bu yeni çözüm üzerinden devam ettirilmektedir. Basit yapısına

rağmen YYA literatürdeki pek çok kombinatoriyal optimizasyon problemine başarıyla uygulanmıştır. Bu problemlere karesel atama problemi (Stütze, 2006), sırt çantası problemi (Avcı ve Topaloglu, 2017), araç rotalama problemleri (Subramanian, Drummond, Bentes, Ochi ve Farias, 2010; Penna, Subramanian ve Ochi, 2013; Michallet, Prins, Amodeo, Yalaoui ve Vitry, 2014; Brandão, 2020) ve çizelgeleme problemleri (Dong, Chen, Huang ve Nowak, 2013; Zohali, Naderi, Mohammadi ve Roshanaei, 2019) örnek olarak verilebilir. YYA çalışmaları ile ilgili detaylı bilgi edinebilmek için Lourenço, Martin ve Stütze (2019) tarafından yapılan derleme çalışması incelenebilir. İlgili literatür incelendiğinde ele alınan İA-TYP için YYA'nın henüz uygulanmadığı görülmektedir. Bu çalışmanın amacı literatürdeki bu eksikliği gidermektir. Önerilen algoritmanın performansı, ilgili literatürde yer alan İA-TYP kıyaslama örnekleri kullanılarak test edilmiştir. Elde edilen sonuçlar YYA'nın İA-TYP'yi çözmedeki etkinliğini ortaya koymaktadır.

İkinci bölümde ele alınan problemin tanıtımı ve modellenmesi, bilimsel yazın taraması beraber sunulmuştur. Üçüncü bölümde geliştirilen YYA'nın özellikleri anlatılmaktadır. Dördüncü bölümde sayısal deney sonuçları ve son olarak beşinci bölümde sonuçlara ait değerlendirmeler yer almaktadır.

2. Problem Tanıtımı ve Bilimsel Yayın Taraması

TYP literatürde genellikle Karesel Atama Problemi (KAP) olarak modellenmektedir. $N = \{1, \dots, n\}$ ve $M = \{1, \dots, m\}$ sırasıyla kaynak ve konum kümelerini ifade etsin. Çok Amaçlı TYP (ÇA-TYP) için KAP formülasyonu aşağıdaki gibi oluşturulabilir:

$$\min \sum_{i \in N} \sum_{j \in M} \sum_{k \in N \setminus \{i\}} \sum_{l \in M \setminus \{j\}} A_{ijkl} X_{ij} X_{kl} \quad (1)$$

Öyle ki;

$$\sum_{i \in N} X_{ij} = 1 \quad j \in M \quad (2)$$

$$\sum_{j \in M} X_{ij} = 1 \quad i \in N \quad (3)$$

$$X_{ij} \in \{0,1\} \quad i \in N, j \in M \quad (4)$$

Burada X_{ij} ikili karar değişkeninin değeri, $i \in N$ kaynağının $j \in M$ konumuna atanması durumunda

1'e, aksi takdirde 0'a eşittir. A_{ijkl} parametresi ise i kaynağının j konumuna ve k kaynağının l konumuna atanması durumunda ortaya çıkan maliyeti ifade etmektedir. Amaç fonksiyonu (1) toplam maliyeti minimize etmektedir. Eşitlik (2) ve (3) sırasıyla her konuma yalnızca bir kaynağın ve her kaynağın yalnızca bir konuma atanmasını sağlamaktadır. Eşitlik (4) ise karar değişkenlerinin yapısını tanımlamaktadır. KAP'ın NP-Zor bir problem olduğu Sahni ve Gonzales (1976) tarafından yapılan çalışmada gösterilmiştir. n adet kaynak ve konum içeren bir KAP örneği için $n!$ adet alternatif çözüm vardır. Yani kaynak sayısı arttıkça problemin boyutu üstel olarak artmaktadır.

Literatürde ÇA-TYP'nin ele alındığı pek çok çalışma yer almaktadır. Bu çalışmaların büyük bir bölümünde yukarıda bahsedilen nitel ve nicel amaçlar dikkate alınmıştır. ÇA-TYP ilk olarak Rosenblatt (1979) tarafından literatüre sunulmuştur. Bu çalışmada A_{ijkl} parametresi, $A_{ijkl} = w_1 f_{ik} d_{jl} - w_2 cr_{ik}$ biçiminde ifade edilmiştir. Burada f_{ik} ve cr_{ik} , sırasıyla i ve k kaynakları arasındaki iş akışı miktarını ve yakınlık derecelendirme puanını temsil etmektedir. d_{jl} , j ve l konumları arasındaki mesafeyi ve son olarak w_1 ve w_2 amaç ağırlıklarını ifade etmektedir. Rosenblatt (1979) problemin çözümü için bir grafiksel yaklaşım geliştirmiştir. Aynı problem tanımı Dutta ve Sahu (1982) tarafından da kullanılmıştır. Yazarlar problem için bir iyileştirme sezgiseli önermiştir. Fortenberry ve Cox (1985) ÇA-TYP için bir çarpımsal model geliştirmiştir. Bu modelde A_{ijkl} parametresinin değeri cr_{ik} , f_{ik} ve d_{jl} değerlerinin birbirleriyle çarpımına eşittir. Amaç ağırlıkları bu modelde elenmiştir. Urban (1989) nitel ve nicel terimlerin birleştirildiği bir model geliştirmiştir. Bu modelde nitel amaç bir ağırlıkla çarpılarak nicel amaca eklenmektedir. Bu eklemeli yaklaşım daha sonra Khare, Khare ve Neema (1988) tarafından da kullanılmıştır. Chen ve Sha (1999) ÇA-TYP kapsamında baskınlık endeksi ve üstün olma olasılığı kavramlarını çözümlerin kalitesini belirlemede kullanmıştır. Aynı yaklaşım Sha ve Chen (2001) tarafından yapılan çalışmada da uygulanmıştır. ÇA-TYP ile ilgili Malakooti ve Tsurushima (1989), Suresh ve Sahu (1993), Chen ve Sha (2005), Mohamadi, Ebrahimnejad ve Tavakkoli-Moghaddam (2018) ve Shojaie ve Bariran (2020) tarafından yapılan çalışmalar da ayrıca incelenebilir. Malakooti ve Tsurushima (1989) ÇA-TYP için çok kriterli karar verme tabanlı bir uzman sistem önermiştir. Önerilen sistem iki aşamadan oluşmaktadır: birinci aşamada bir kural ve kısıt

kümesine bağlı olarak bir yerleşim üretilmektedir, bu yerleşim ikinci aşamada karar vericiyle etkileşim halinde bulunularak iyileştirilmektedir. Suresh ve Sahu (1993) ÇA-TYP için bir Tavlama Benzetimi (TB) algoritması önermiştir. Chen ve Sha (2005) problem için 5 aşamalı bir sezgisel yaklaşım önermiştir. Burada ilk aşamada temel bir yerleşim planı rastgele amaç ağırlıkları ile üretilmektedir. İkinci aşamada amaçlar arası eşleştirilmiş karşılaştırma matrisi bir prosedür yardımıyla oluşturulmaktadır. Oluşturulan matrisin tutarlı olup olmadığı üçüncü aşamada tespit edilmekte ve tutarsız olması durumunda dördüncü aşamada tutarlı hale dönüştürülmektedir. Son olarak beşinci aşamada tercih edilen amaç ağırlıkları türetilmekte ve buna göre yerleşim planı oluşturulmaktadır. Toplam malzeme taşıma maliyetinin ve ölü alanların eş zamanlı olarak minimize edildiği bir İA-TYP Mohamadi, Ebrahimnejad ve Tavakkoli-Moghaddam (2018) ve Shojaie ve Bariran (2020) tarafından çalışılmıştır. İlgili problem için ilk çalışmada 3 adet metasezgisel algoritma önermiştir. Önerilen yöntemler sırasıyla genetik algoritma, parçacık sürüsü optimizasyonu ve paralel TB tabanlı algoritmalar. İkinci çalışmada ise parçacık sürüsü optimizasyonu ve paralel TB tabanlı iki algoritma problemin çözümü için önerilmiştir.

Yukarıda bahsedilen ÇA-TYP modellerinin çoğu birbirine benzemektedir ve aralarındaki tek farklılık maliyet terimi A_{ijkl} ile nitel ve nicel ölçeklerin ilişkisinin ifade edilmesinde ortaya çıkmaktadır. Chen ve Sha (1999) tarafından da belirtildiği gibi, bu modellerde iki temel problem bulunmaktadır. Bu problemlerden birincisi tüm amaçların aynı ölçekte temsil edilememesi durumudur, ikinci problem ise amaçlar için kullanılan ölçü birimlerinin kıyaslanamaz durumda olmasıdır. Bu sebeplerden dolayı Harmonosky ve Tothoro (1992) tüm amaçları birleştirmeden önce normalize eden bir yaklaşım önermiştir. Bir amacın normalize edilmesi için o amaçla ilgili her ilişki değerinin ilgili tüm ilişki değerlerinin toplamına bölünmesi gerekmektedir. Bu işlem aşağıdaki gibi gerçekleştirilebilir:

$$T_{ikm} = S_{ikm} / \sum_{i \in N} \sum_{k \in N \setminus \{i\}} S_{ikm} \quad (5)$$

Burada S_{ikm} , m amacı için i ve k kaynakları arasındaki ilişkiyi, T_{ikm} ise normalize edilmiş ilişki değerini temsil etmektedir. T_{ikm} değerlerini içeren ÇA-TYP matematiksel modelinin amaç fonksiyonu Eşitlik (6)'da gösterilmektedir. Amaç fonksiyonu olarak Eşitlik (1) yerine Eşitlik (6)'nın kullanıldığı

ÇA-TYP matematiksel modeli (2) – (4) eşitliklerinde belirtilen kısıtlardan oluşmaktadır.

$$\min \sum_{i \in N} \sum_{j \in M} \sum_{k \in N \setminus \{i\}} \sum_{l \in M \setminus \{j\}} \sum_{m=1}^P w_m T_{ikm} d_{jl} X_{ij} X_{kl} \quad (6)$$

Bu modelde P amaç sayısını temsil etmektedir. Bu çalışmada yukarıdaki model kullanılmıştır. Yani, öncelikle her iki amaç normalize edilmiş, ardından İA-TYP tek amaçlı bir problem gibi çözülmüştür. Aynı model Şahin (2011) ve Matai, Singh ve Mittal (2013) tarafından yapılan çalışmalarda da kullanılmıştır. İki çalışmada da yazarlar problem için Tavlama Benzetimi (TB) tabanlı sezgisel çözüm yöntemleri önermiştir.

Amaç fonksiyonunun hesaplama adımları kaynak sayısı 5 olan bir örnek problem üzerinde gösterilmiştir. Nitel ve nicel olmak üzere iki farklı amaca sahip örnek problem Tablo 1’de gösterilmektedir. Tabloda nitel ilişki değerleri A ve X harfleri arasında değişen yakınlık derecelendirme puanlarından oluşmaktadır. Nicel ilişki değerleri ise

1 – 10 arasında değişen kaynaklar arası iş akışı değerleriyle temsil edilmektedir. A harfi iki kaynak arasındaki en yüksek yakınlık derecelendirme puanını temsil ederken, yakın olması istenmeyen kaynaklar arasındaki ilişki X harfiyle gösterilmiştir. Nitel açıdan yakınlıkları önemli olmayan iki kaynak arasındaki ilişki değeri U harfiyle gösterilmiştir. Nitel ilişki değerleri Dutta ve Sahu (1982) tarafından önerildiği gibi -1 (X) ve 4 (A) arasında değişecek şekilde sayısal hale getirilmiştir. Tablo 2’de ilişki değerleri sayısal olarak verilmiştir.

Tablo 2’deki değerler kullanılarak oluşturulan normalize edilmiş ilişki değerleri Tablo 3’te verilmiştir. Son olarak nitel ve nicel amaç ağırlıklarının sırasıyla 0,3 ve 0,7 olduğu durumda kaynaklar arasında oluşan bileşik ilişki değerleri Tablo 4’te gösterilmektedir.

Tablo 1
Örnek Problem

	Nitел					Nicel					
	1	2	3	4	5	1	2	3	4	5	
1		A	O	X	I	1		4	2	5	3
2	A		E	U	O	2	1		6	3	7
3	O	E		U	I	3	5	2		9	5
4	X	U	U		A	4	7	2	6		10
5	I	O	I	A		5	9	7	4	2	

Tablo 2
Nitел ve Nicel İlişki Değerlerinin Sayısal Olarak Gösterimi

	Nitел					Nicel					
	1	2	3	4	5	1	2	3	4	5	
1		4	1	-1	2	1		4	2	5	3
2	4		3	0	1	2	1		6	3	7
3	1	3		0	2	3	5	2		9	5
4	-1	0	0		4	4	7	2	6		10
5	2	1	2	4		5	9	7	4	2	

Tablo 3
Normalize Edilmiş İlişki Değerleri

Nitel					Nicel				
1	2	3	4	5	1	2	3	4	5
1	0,1250	0,0313	-0,0313	0,0625	1	0,0404	0,0202	0,0505	0,0303
2	0,1250	0,0938	0,0000	0,0313	2	0,0101	0,0606	0,0303	0,0707
3	0,0313	0,0938	0,0000	0,0625	3	0,0505	0,0202	0,0909	0,0505
4	-0,0313	0,0000	0,0000	0,1250	4	0,0707	0,0202	0,0606	0,1010
5	0,0625	0,0313	0,0625	0,1250	5	0,0909	0,0707	0,0404	0,0202

Tablo 4
Bileşik İlişki Değerleri

Nitel ağırlık = 0,3 – Nicel ağırlık = 0,7					
	1	2	3	4	5
1		0,0658	0,0235	0,0260	0,0400
2	0,0446		0,0705	0,0212	0,0589
3	0,0447	0,0423		0,0636	0,0541
4	0,0401	0,0141	0,0424		0,1082
5	0,0824	0,0589	0,0470	0,0516	

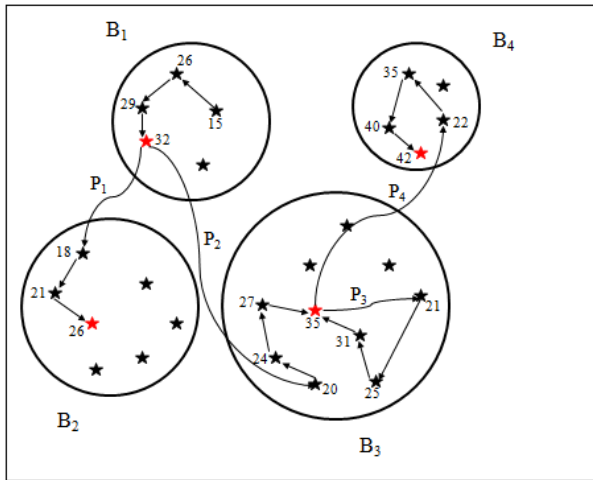
3. Yinelemeli Yerel Arama Algoritması

Bu bölümde İA-TYP'nin çözümü için önerilen Yinelemeli Yerel Arama (YYA) algoritması tanıtılmıştır. YYA bir metasezgisel algoritmadır. Metasezgiseller, çözüm uzayını etkili bir biçimde araştırmak için temel sezgisel metotları bütünleştiren algoritmik yapılardır. Metasezgisel yöntemlerin uygulanmasıyla, çözüm uzayının verimli bir şekilde etüt edilerek makul bir hesaplama süresi içinde optimum veya optimuma yakın çözümlerin elde edilmesi hedeflenmektedir. Metasezgisel algoritmaların verimliliğini etkileyen en önemli faktörlerden biri, çeşitlendirme ve yoğunlaştırma arasındaki dengedir. Çeşitlendirme terimi, algoritmanın çözüm uzayındaki farklı arama bölgelerini keşfetme yeteneği olarak tanımlanabilirken, yoğunlaştırma terimi, keşfedilen arama bölgelerinden yüksek kaliteli çözümler elde etmekle ilgilidir.

YYA'da yoğunlaştırma ve çeşitlendirme sırasıyla yerel arama ve saptırma mekanizmalarıyla sağlanmaktadır. YYA bir başlangıç çözümü üreterek başlamaktadır. Daha sonra yerel arama uygulanarak üretilen çözüm iyileştirilmektedir. YYA'nın her yinelemesinde öncelikle mevcut çözüme saptırma uygulanarak bir saptırılmış çözüm bulunmaktadır. Daha sonra saptırılmış çözüm yerel arama uygulanarak iyileştirilmektedir. Elde edilen

çözümün belirlenen kriterlere göre kabul edilebilir olması durumunda mevcut çözüm güncellenmektedir. Bu prosedür, sonlandırma koşulu sağlanana kadar tekrar edilmektedir.

Bir maksimizasyon problemi için bir YYA algoritmasının arama mekanizması Şekil 1'de örneklendirilmektedir. Burada birinci çekim havzası B₁'de yerel optimum noktanın amaç fonksiyonu değeri 32'dir. İlk saptırma P₁'in uygulanmasıyla arama başka bir çekim havzası olan B₂'ye yönelir, ancak burada uygulanan yerel arama sonrası bulunan yerel optimum (amaç fonksiyonu değeri 26) öncekinden daha iyi değildir. İkinci saptırma ile B₃ çekim havzasında bulunan ve amaç fonksiyonu değeri 20 olan bir çözüm elde edilir. Burada yapılan yerel arama ile elde edilen yerel optimum çözümün amaç fonksiyonu değeri önceki yinelemelerde elde edilen en iyi çözümden daha iyidir. Bu sebeple mevcut çözüm güncellenir (amaç fonksiyonu değeri 35). Üçüncü saptırma hareketi ile elde edilen çözüm aynı çekim havzasında bulunmaktadır. B₃ sonrası, YYA P₄ saptırması ile yeni bir çekim havzasına ulaşmakta ve burada amaç fonksiyonu değeri 42 olan yeni bir yerel optimum çözüm elde etmektedir.



Şekil 1. YYA'nın Arama Mekanizması Gösterimi

Önerilen YYA'nın uygulama adımları Algoritma 1'de gösterilmektedir. YYA, $maxF$ ve $maxSL$ olmak üzere iki parametre içermektedir. $maxF$ iyileştirme olmadan art arda maksimum yineleme sayısını, $maxSL$ ise maksimum saptırma büyüklüğünü göstermektedir. Algoritma bir başlangıç çözümünün rastgele oluşturulmasıyla başlamaktadır. Daha sonra elde edilen çözüm yerel arama prosedürü YA'nın uygulanmasıyla iyileştirilmektedir. Algoritmamızda yerel arama, rastgele komşuluk sıralamalı (RKS) bir Değişken Komşuluklu İniş (DKİ) (Mladenović ve Hansen, 1997) algoritmasına dayanan bir prosedüre sahiptir. Klasik DKİ'de arama sabit bir komşuluk sırası gözetilerek gerçekleştirilmektedir. RKS stratejisi uygulandığında bu komşuluk sırası rastgele olmaktadır. Penna, Subramanian ve Ochi (2013) tarafından yapılan çalışmada da gösterildiği gibi RKS'nin uygulanması DKİ performansına olumlu etki yapabilmektedir. Algoritmanın her yinelemesi (Satır 6–20) mevcut çözüme Saptır prosedürünün uygulanmasıyla başlamaktadır. *Saptır* prosedüründe mevcut çözüme sL sayıda rastgele hareket uygulanarak yeni bir çözüm (x_s) elde edilmektedir. Ardından x_s çözümüne YA uygulanarak x_a aday çözümü elde edilmektedir. Elde edilen aday çözümün mevcut çözüme eşit olması durumu uygulanan saptırma hareketinin yetersizliğini göstermektedir. Bu sebeple böyle bir durum olduğunda *Saptır* prosedüründe uygulanan hareket sayısı 1 artırılmaktadır. Aksi durumda x_a 'nın mevcut çözümden iyi olup olmadığı kontrol edilmektedir (Satır 12). $f(x_a) < f(x)$ ise, mevcut çözüm güncellenir ve art arda başarısız yineleme sayısı f ve saptırma büyüklüğü sL değerleri sırasıyla 0 ve 1'e eşitlenir. Aksi takdirde f sayacı 1

artırılır. YYA, f sayacı $maxF$ sayısını geçtiğinde veya sL değeri $maxSL$ 'den büyük olduğunda sonlandırılır. Takip eden alt bölümde YA detaylı bir şekilde açıklanacaktır.

Algoritma 1: Geliştirilen YYA'nın temel adımları

Girdi: Probleme ilişkin veriler, $maxF$, $maxSL$

Çıktı: En iyi çözüm

```

1  Rastgele bir başlangıç çözümü  $x$  oluştur
2   $x' \leftarrow YA(x)$ 
3   $x \leftarrow x'$ 
4   $f \leftarrow 0$ 
5   $sL \leftarrow 1$ 
6  While  $f \leq maxF$  &  $sL \leq maxSL$ 
7       $x_s \leftarrow Saptır(x, sL)$ 
8       $x_a \leftarrow YA(x_s)$ 
9      If  $x = x_a$ 
10          $sL \leftarrow sL + 1$ 
11     Else
12         If  $f(x_a) < f(x)$ 
13              $x \leftarrow x_a$ 
14              $sL \leftarrow 1$ 
15              $f \leftarrow 0$ 
16         Else
17              $f \leftarrow f + 1$ 
18         End If
19     End If
20 End While

```

3.1 Yerel Arama Prosedürü

YA prosedürünün temel adımları Algoritma 2'de verilmiştir. İlgili prosedür tüm komşuluk yapılarından oluşan L kümesinin oluşturulmasıyla başlamaktadır. Prosedürdeki her yinelemede (Satır 2–11) öncelikle L kümesinden bir k komşuluk yapısı rastgele seçilmektedir. Daha sonra seçilen komşuluk yapısının mevcut çözüm x' e uygulanmasıyla bulunabilecek en iyi çözüm x' belirlenir. Komşu çözüm x' eğer mevcut çözümden daha iyiyse yeni mevcut çözüm olarak kabul edilir ve L kümesi başlangıç durumuna getirilir. Aksi takdirde k komşuluk yapısı L kümesinden çıkarılır. Aynı işlemler L kümesi boş küme olana kadar devam eder.

Algoritma 2: YA'nın temel adımları**Girdi:** x **Çıktı:** x

```

1  Komşuluk yapısı kümesini ( $L$ ) başlangıç
   durumuna getir
2  While  $L \neq \emptyset$ 
3      Rastgele bir komşuluk yapısı  $k \in L$  seç
4       $k$  komşuluk yapısı ile bulunabilecek en iyi
       çözümlü ( $x'$ ) belirle
5      If  $f(x') < f(x)$ 
6           $x \leftarrow x'$ 
7           $L$  kümesini başlangıç durumuna getir
8      Else
9           $L \leftarrow L \setminus \{k\}$ 
10     End If
11 End While

```

3.1.2 Komşuluk Yapıları

YA prosedüründe *İkili Takas*, *Üçlü Takas* ve *Yerleştir* olmak üzere 3 adet komşuluk yapısı kullanılmaktadır. Komşuluk yapılarının gösterimi için kullanılacak örneklerde kaynaklar Şekil 2'de belirtilen konumlara atanacaktır. Komşuluk yapılarının açıklamaları aşağıdaki gibidir.

1	2	3
4	5	6
7	8	9

Şekil 2. Komşuluk Gösteriminde Kullanılan Konumlar

❖ *İkili Takas*: Bu komşuluk yapısında mevcut çözümdeki iki farklı konumda bulunan kaynaklar birbirleriyle takas edilir. Şekil 3'te 1. konumda bulunan kaynak 9 ve 4. konumda bulunan kaynak 4 birbiriyle takas edilmiştir.

9	7	1
4	2	3
5	6	8

İkili Takas →

4	7	1
9	2	3
5	6	8

Şekil 3. İkili Takas Komşuluk Yapısı Uygulaması Örneği

❖ *Üçlü Takas*: Bu komşuluk yapısında 3 farklı konumda bulunan kaynaklar birbirleriyle takas edilir. Şekil 4'te birinci, beşinci ve dokuzuncu konumda bulunan kaynakların birbirleriyle yerleri değiştirilmiştir. Burada her üçlü konum kombinasyonu için mevcut alternatif hamlelerden yalnızca biri rastgele seçilerek uygulanmaktadır.

9	7	1
4	2	3
5	6	8

Üçlü Takas →

8	7	1
4	9	3
5	6	2

Şekil 4. Üçlü Takas Komşuluk Yapısı Uygulaması Örneği

❖ *Yerleştir*: Bu yapıda bir kaynak mevcut konumundan çıkarılıp ardışık numaralı konumlarda bulunan iki kaynak arasına yerleştirilmektedir. Şekil 5'te altıncı konumda bulunan kaynak 3 mevcut konumundan çıkarılmış ve sekizinci ve dokuzuncu konumlarda bulunan kaynak 6 ve 8 arasına yerleştirilmiştir. Yeni çözüm incelendiğinde konum 1 - 5 ve 9'da bulunan kaynakların yerlerinde bir değişiklik olmamıştır. Buna karşılık kaynak 3'ün konum 6'dan çıkarılmasıyla yedinci ve sekizinci konumda bulunan kaynaklar sırasıyla altıncı ve yedinci konuma geçmişlerdir ve kaynak 3 konum 8'e atanmıştır.

9	7	1
4	2	3
5	6	8

Yerleştir →

9	7	1
4	2	5
6	3	8

Şekil 5. Yerleştir Komşuluk Yapısı Uygulaması Örneği

3.2 Saptırma Prosedürü

Önerilen YYA'da uygulanan saptırma prosedürü *Saptır*'ın temel adımları Algoritma 3'te verilmiştir. *Saptır* prosedürü sL sayıda yinelemeden oluşmaktadır. Her yinelemede bir önceki alt bölümde anlatılan komşuluk yapılarından biri

rastgele seçilmektedir ve seçilen komşuluk yapısı kullanılarak bir rastgele çözüm oluşturulmaktadır. Prosedürün ilk yinelemesinde seçilen komşuluk yapısı mevcut çözüme uygulanmaktadır. Eğer sL değeri 1'e eşitse prosedür burada sonlandırılmaktadır. Aksi takdirde 2'den sL 'ye kadar olan her yinelemede bir önceki yinelemenin sonucunda elde edilen çözüm üzerinden yeni bir çözüm üretilmektedir.

Algoritma 3: *Saptır* prosedürünün temel adımları

Girdi: x, sL

Çıktı: x_s

```

1   $x_s^0 \leftarrow x$ 
2  For  $i \leftarrow 1 \dots sL$ 
3      Rastgele bir komşuluk yapısı ( $k$ ) seç
4       $k$  komşuluk yapısını  $x_s^{i-1}$  çözümüne rastgele
        uygulayarak  $x_s^i$  çözümünü oluştur
5  End For
6   $x_s \leftarrow x_s^{sL}$ 

```

Bu çalışmada araştırma ve yayın etiğine uyulmuştur.

4. Sayısal Deneysel Sonuçları

Bu bölümde önerilen YYA'nın performansını ölçmek için yapılan sayısal deneylerin sonuçları verilmektedir. YYA MATLAB'da kodlanmıştır. Tüm sayısal deneyler 2.60GHz Intel Core i7-10750H işlemci ve 16 GB RAM özelliklerine sahip bir kişisel bilgisayarda gerçekleştirilmiştir. Algoritmanın performansı ilk olarak Chen ve Sha (1999) tarafından sunulan 4 adet veri kümesi üzerinde test edilmiştir. Bu 4 adet veri kümesinde kaynak/konum sayısı sırasıyla 8, 12, 15 ve 20'dir. Ayrıca bu çalışma kapsamında kaynak sayısı 40 olan yeni bir veri kümesi türetilmiştir. Türetilen veri kümesinde kaynaklar Şekil 6'da gösterilen (5×8) dikdörtgen üzerinde belirtilen konumlara atanacaktır. Konumlar arası uzaklıklar dik doğrusal olarak alınmıştır ve her konumun genişliği 1 birimdir. Kaynaklar arasındaki yük miktarları [1, 20] aralığında rastgele türetilmiştir. Nitel ilişki değerleri ise bir önceki örnekte gösterildiği gibi A ve X harfleri arasında rastgele türetilmiştir. Talep olması durumunda türetilen probleme ait veriler yazar tarafından sağlanacaktır. Her bir veri kümesi için farklı amaç ağırlığı kombinasyonlarının olduğu 11 problem örneği dikkate alınmıştır. Sonuç olarak, toplamda $11 \times 5 = 55$ farklı problem örneği için YYA çalıştırılmıştır. Burada kaynak sayısının 8, 12, 15, 20 ve 40 olduğu veri kümeleri sırasıyla veri kümesi 1,

2, 3, 4 ve 5 olarak adlandırılacaktır.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

Şekil 6. Türetilen Veri Kümesine İlişkin Konumlar

4.1 Chen ve Sha (1999) Tarafından Sunulan Veri Kümeleri Üzerinde Elde Edilen Sonuçlar

Geliştirilen YYA, Chen ve Sha (1999) tarafından sunulan veri kümeleri üzerinde, Şahin (2011) tarafından önerilen Tavlama Benzetimi (TB) ve Matai, Singh ve Mittal (2013) tarafından geliştirilen Değiştirilmiş Tavlama Benzetimi (DTB) algoritmalarıyla karşılaştırılmıştır. Her iki algoritmanın da 44 problem örneği için ürettiği çözümler (yerleşim planları) ilgili makalelerde rapor edilmiştir. YYA'nın ürettiği çözümlerin kalitesi, TB ve DTB algoritmalarının ürettikleri ve bahsi geçen çalışmalarda rapor edilmiş çözümlerin kalitesi baz alınarak değerlendirilmiştir.

Şahin (2011) ve Matai, Singh ve Mittal (2013) tarafından üretilen çözümler için rapor edilen amaç fonksiyonu değerleri incelendiğinde ilgili değerlerin hesaplanmasında uygulama farklılıkları olduğu anlaşılmaktadır. Bu konuyla ilgili olarak her iki çalışmanın da sorumlu yazarlarıyla iletişime geçilmiştir. Şahin (2011)'in yazarı çalışmada normalize edilmiş ilişki değerlerini virgülden sonra üçüncü basamağa yuvarladığını belirtmiştir. Matai, Singh ve Mittal (2013)'in sorumlu yazarından ise herhangi bir cevap alınamamıştır. Fakat sonuçlar incelendiğinde bu çalışmada normalize edilmiş ilişki değerlerinin yuvarlama işlemi yapılmadan kullanıldığı anlaşılmaktadır. Dolayısıyla uygun bir karşılaştırma yapabilmek için YYA her bir problem örneği için iki farklı normalizasyon uygulanarak çalıştırılmıştır.

Önerilen YYA'nın $maxF$ ve $maxSL$ olmak üzere bitirme kriterinde yer alan iki parametresi bulunmaktadır. Yapılan ön deneyler sonucunda, $maxF$ ve $maxSL$ parametrelerinin değerleri sırasıyla 200 ve 20'ye sabitlemiştir. Geliştirilen

algoritma her bir problem örneği için 10 defa çalıştırılmıştır. Chen ve Sha (1999) tarafından sunulan veri kümeleri üzerinde gerçekleştirilen sayısal deneylerin sonuçları Tablo 5 - 8'de yer almaktadır. Her tabloda aynı kaynak sayısına sahip problem örnekleri için elde edilen sonuçlar verilmektedir. Ayrıca Ekler bölümünde, Ek Tablo 1 - 8'de her bir problem örneği için YYA tarafından üretilen en iyi yerleşim gösterilmektedir. Tablo 5 - 8'de, ilk iki sütun her problem için amaç ağırlıklarını vermektedir. İlgili tablolarda YYA, TB ve DTB ile ayrı ayrı karşılaştırılmaktadır. Burada Şahin (2011) tarafından uygulanan normalizasyon 1. tip, Matai, Singh ve Mittal (2013) tarafından uygulanan (uygulandığı düşünülen) normalizasyon 2. tip normalizasyon olarak adlandırılmaktadır. Bahsi geçen tablolarda "A. D.", amaç fonksiyonu değerinin kısaltmasıdır. Yine aynı tablo grubunda YYA için 10 tekrar sonucu elde edilen en iyi ve ortalama amaç fonksiyonu değerleri gösterilmektedir. Kalın sayılar ilgili problem örneği için çözüm yönteminin bilinen en iyi çözümü ürettiğini göstermektedir. Ayrıca bazı problem örnekleri için Matai, Singh ve Mittal (2013) tarafından DTB için rapor edilen amaç fonksiyonu değeriyle, sunulan yerleşim planının amaç fonksiyonu değeri birbirini tutmamaktadır. Bir problem örneğinde ise sunulan yerleşim planı eksiktir (Kaynak 6 çözümde bulunmamaktadır.). İlgili makalenin sorumlu yazarından bu sorunlarla ilgili olarak da bilgi talep edilmiştir. Fakat daha önce de belirtildiği gibi cevap alınamamıştır.

Kaynak sayısının 8, 12 ve 15 olduğu problem örnekleri için elde edilen sonuçlar sırasıyla Tablo 5, 6 ve 7'de gösterilmektedir. 1. tip normalizasyon ile elde edilen sonuçlar incelendiğinde YYA ve TB'nin 33 problemin tümü için ürettikleri en iyi çözümler aynı kalitededir. 2. tip normalizasyon ile elde edilen sonuçlarda da benzer bir durum gözlemlenmektedir. Burada 31 problem örneği için YYA ve DTB'nin ürettikleri en iyi çözümlerin kalitesi birbirine denktir. Ayrıca, YYA 8 ve 12 kaynağa sahip her problem örneği için yapılan tüm tekrarlarla aynı kaliteye sahip çözümler üretmiştir. 15 kaynağa sahip problemlerde ise 1. ve 2. tip normalizasyon için en iyi ve ortalama amaç fonksiyonu değerleri arasındaki ortalama farkın sırasıyla %0,10 ve %0,15 olduğu tespit edilmiştir.

Kaynak sayısının 20 olduğu problem örnekleri için elde edilen sonuçlar Tablo 8'de verilmektedir. Sonuçlar incelendiğinde, YYA'nın tüm problemler için en iyi çözümleri bulabildiği görülmektedir. Ayrıca, önerilen algoritma 1. tip (2. tip) normalizasyon ile 3 problemde (5 problemde) TB

(DTB) ile bulunan en iyi çözümü iyileştirmeyi başarmıştır. Öte yandan YYA'nın en iyi ve ortalama amaç fonksiyonu değerleri incelendiğinde, iki değer arasındaki farkın her iki tip normalizasyon için de ortalama %0,02 olduğu görülmektedir. YYA'nın ortalama ve en iyi çözüm kaliteleri arasındaki farklar önerilen yöntemin sağlamlığını ortaya koymaktadır.

4.1 Türetilen Veri Kümesi Üzerinde Elde Edilen Sonuçlar

Önerilen YYA'nın performansını türetilen veri kümesi üzerinde başka bir sezgisel algoritmayla karşılaştırmak amacıyla Şahin (2011) tarafından önerilen TB algoritması MATLAB ortamında kodlanmıştır. TB'nin bu versiyonu TB' ile gösterilmektedir. Algoritmanın başlangıç sıcaklığı (T_{in}), soğutma hızı (α), maksimum yineleme sayısı (el_{max}) ve aynı sıcaklık seviyesindeki deneme sayısı ($NIET$) olmak üzere dört parametresi bulunmaktadır. T_{in} parametresi her çalıştırmada başlangıç çözümü amaç fonksiyonu değerine eşitlenmiştir. Yapılan ön deneyler sonucunda α , el_{max} ve $NIET$ parametreleri sırasıyla 0,999, 9000 ve 5 değerlerine sabitlenmiştir.

Elde edilen sonuçlar Tablo 9'da gösterilmektedir. Ayrıca Ekler bölümünde, Ek Tablo 9'da YYA'nın her problem örneği için ürettiği en iyi çözüm sunulmaktadır. Sonuçlara göre YYA ve TB' 6 problem örneği için aynı kalitede en iyi çözüm üretmiştir. 5 örnekte ise YYA'nın en iyi çözümü TB' ile elde edilen en iyi çözümden daha iyidir. En iyi ve ortalama amaç fonksiyonu değerleri arasındaki ortalama fark YYA için %0,117, TB' içinse %0,129 bulunmuştur. Bu sonuçtan yola çıkarak her iki yöntemin de sağlam olduğu söylenebilir. Yine ortalama hesaplama süreleri açısından da iki yöntem arasında büyük bir fark bulunmamaktadır. TB' ve YYA'nın ortalama hesaplama süreleri sırasıyla 12,51 ve 14,71 saniyedir. Tüm sonuçlar beraber incelendiğinde 11 örneğin 5'inde daha kaliteli en iyi çözümler üretebilmesi nedeniyle önerilen YYA'nın İA-TYP'nin çözümünde daha etkili bir yöntem olduğu görülmektedir.

Tablo 5
Veri Kümesi 1 için Elde Edilen Sonuçlar

		1. tip normalizasyon ile elde edilen sonuçlar			2. tip normalizasyon ile elde edilen sonuçlar		
		TB	YYA		DTB	YYA	
w_1	w_2	Amaç D.	En iyi A. D.	Ort. A. D.	Amaç D.	En iyi A. D.	Ort. A. D.
1	0	1,7850	1,7850	1,7850	1,7946	1,7946	1,7946
0,9	0,1	1,8124	1,8124	1,8124	1,8223	1,8223	1,8223
0,8	0,2	1,8104	1,8104	1,8104	1,8196*	1,8196	1,8196
0,7	0,3	1,7669	1,7669	1,7669	1,7750	1,7750	1,7750
0,6	0,4	1,7034	1,7034	1,7034	1,7098	1,7098	1,7098
0,5	0,5	1,6235	1,6235	1,6235	1,6295	1,6295	1,6295
0,4	0,6	1,5420	1,5420	1,5420	1,5473	1,5473	1,5473
0,3	0,7	1,4605	1,4605	1,4605	1,4652	1,4652	1,4652
0,2	0,8	1,3790	1,3790	1,3790	1,3830	1,3830	1,3830
0,1	0,9	1,2975	1,2975	1,2975	1,3009	1,3009	1,3009
0	1	1,2160	1,2160	1,2160	1,2188	1,2188	1,2188

* DTB için rapor edilen amaç fonksiyonu değeriyle, sunulan yerleşim planının amaç fonksiyonu değeri birbirini tutmamaktadır.

Tablo 6
Veri Kümesi 2 için Elde Edilen Sonuçlar

		1. tip normalizasyon ile elde edilen sonuçlar			2. tip normalizasyon ile elde edilen sonuçlar		
		TB	YYA		DTB	YYA	
w_1	w_2	Amaç D.	En iyi A. D.	Ort. A. D.	Amaç D.	En iyi A. D.	Ort. A. D.
1	0	2,1050	2,1050	2,1050	2,1075	2,1075	2,1075
0,9	0,1	2,1235	2,1235	2,1235	2,1226	2,1226	2,1226
0,8	0,2	2,0988	2,0988	2,0988	2,0969	2,0969	2,0969
0,7	0,3	2,0489	2,0489	2,0489	2,0448	2,0448	2,0448
0,6	0,4	1,9982	1,9982	1,9982	1,9928	1,9928	1,9928
0,5	0,5	1,9410	1,9410	1,9410	1,9353	1,9353	1,9353
0,4	0,6	1,8748	1,8748	1,8748	1,8677	1,8677	1,8677
0,3	0,7	1,8086	1,8086	1,8086	1,8001	1,8001	1,8001
0,2	0,8	1,7386	1,7386	1,7386	1,7288	1,7288	1,7288
0,1	0,9	1,6673	1,6673	1,6673	1,6561	1,6561	1,6561
0	1	1,5960	1,5960	1,5960	1,5833	1,5833	1,5833

Tablo 7
Veri Kümesi 3 için Elde Edilen Sonuçlar

		1. tip normalizasyon ile elde edilen sonuçlar			2. tip normalizasyon ile elde edilen sonuçlar		
		TB	YYA		DTB	YYA	
w_1	w_2	Amaç D.	En iyi A. D.	Ort. A. D.	Amaç D.	En iyi A. D.	Ort. A. D.
1	0	2,2120	2,2120	2,2200	**	2,2477	2,2477
0,9	0,1	2,2692	2,2692	2,2713	2,3013	2,3013	2,3065
0,8	0,2	2,2896	2,2896	2,2896	2,3153	2,3153	2,3172
0,7	0,3	2,2735	2,2735	2,2752	2,2939	2,2939	2,2993
0,6	0,4	2,2544	2,2544	2,2548	2,2725*	2,2723	2,2736
0,5	0,5	2,2060	2,2060	2,2080	2,2208	2,2208	2,2253
0,4	0,6	2,1408	2,1408	2,1439	2,1526	2,1526	2,1586
0,3	0,7	2,0728	2,0728	2,0732	2,0813	2,0813	2,0819
0,2	0,8	1,9980	1,9980	1,9985	2,0048	2,0048	2,0070
0,1	0,9	1,9190	1,9190	1,9220	1,9224	1,9224	1,9264
0	1	1,8400	1,8400	1,8456	1,8400	1,8400	1,8443

* DTB için rapor edilen amaç fonksiyonu değeriyle, sunulan yerleşim planının amaç fonksiyonu değeri birbirini tutmamaktadır.

** DTB için sunulan yerleşim planı tam değildir.

Tablo 8
Veri Kümesi 4 için Elde Edilen Sonuçlar

		1. tip normalizasyon ile elde edilen sonuçlar			2. tip normalizasyon ile elde edilen sonuçlar		
		TB	YYA		DTB	YYA	
w_1	w_2	Amaç D.	En iyi A. D.	Ort. A. D.	Amaç D.	En iyi A. D.	Ort. A. D.
1	0	2,6360	2,6360	2,6367	2,6361*	2,6348	2,6357
0,9	0,1	2,6703	2,6697	2,6700	2,6768*	2,6727	2,6738
0,8	0,2	2,6696	2,6640	2,6669	2,7134*	2,6720	2,6726
0,7	0,3	2,6415	2,6405	2,6409	2,6497*	2,6489	2,6491
0,6	0,4	2,5946	2,5946	2,5949	2,6078*	2,6078	2,6081
0,5	0,5	2,5355	2,5355	2,5355	2,5541	2,5534	2,5534
0,4	0,6	2,4600	2,4600	2,4601	2,4846*	2,4846	2,4850
0,3	0,7	2,3830	2,3830	2,3831	2,4146*	2,4146	2,4146
0,2	0,8	2,3060	2,3060	2,3061	2,3447	2,3447	2,3448
0,1	0,9	2,2236	2,2236	2,2243	2,2689	2,2689	2,2694
0	1	2,1400	2,1400	2,1401	2,1926*	2,1926	2,1927

* DTB için rapor edilen amaç fonksiyonu değeriyle, sunulan yerleşim planının amaç fonksiyonu değeri birbirini tutmamaktadır.

Tablo 9
Veri Kümesi 5 için Elde Edilen Sonuçlar

w_1	w_2	TB'			YYA		
		En iyi A. D.	Ort. A. D.	Ort. S. (s)	En iyi A. D.	Ort. A. D.	Ort. S. (s)
1	0	3,6760	3,6814	9,70	3,6750	3,6783	13,51
0,9	0,1	3,6847	3,6896	9,84	3,6845	3,6871	15,68
0,8	0,2	3,6860	3,6900	8,00	3,6848	3,6893	12,71
0,7	0,3	3,6678	3,6727	8,14	3,6678	3,6722	15,02
0,6	0,4	3,6292	3,6328	9,63	3,6262	3,6301	14,53
0,5	0,5	3,5785	3,5811	11,27	3,5785	3,5817	14,83
0,4	0,6	3,5208	3,5239	10,60	3,5200	3,5226	14,88
0,3	0,7	3,4497	3,4564	13,79	3,4497	3,4544	15,66
0,2	0,8	3,3742	3,3797	16,80	3,3742	3,3815	15,27
0,1	0,9	3,2954	3,2992	18,92	3,2954	3,2979	16,37
0	1	3,2140	3,2194	20,86	3,2140	3,2198	13,32

6. Sonuçlar

Bu çalışmada İki Amaçlı Tesis Yerleşimi Problemi (İA-TYP) ele alınmıştır. Bu problemdeki amaçlar kaynaklar arasındaki toplam yakınlık derecelendirme puanının maksimize edilmesi ve toplam malzeme taşıma maliyetinin en aza indirgenmesidir. Problemin çözümü için bir Yinelemeli Yerel Arama (YYA) algoritması önerilmiştir. Önerilen algoritmada yerel arama prosedürü rastgele komşuluk sıralamalı bir Değişken Komşuluklu İniş algoritmasından oluşmaktadır. Ayrıca YYA'da uygulanan saptırma mekanizması çözüm uzayının yapısına göre uyarlanabilir bir yapıya sahiptir. Geliştirilen YYA'nın performansı 55 farklı problem örneği üzerinde test edilmiştir. Elde edilen sonuçlar algoritmanın İA-TYP'yi çözmedeki etkinliğini ortaya koymaktadır. YYA'nın tüm problem örnekleri için en iyi çözümü üretebildiği gözlemlenmiştir.

Çıkar Çatışması

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir.

Kaynaklar

Avcı, M., & Topaloglu, S. (2017). A multi-start iterated local search algorithm for the generalized quadratic multiple knapsack

problem. *Computers & Operations Research*, 83, 54-65. Doi: <https://doi.org/10.1016/j.cor.2017.02.004>

Brandão, J. (2020). A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. *European Journal of Operational Research*, 284(2), 559-571. Doi: <https://doi.org/10.1016/j.ejor.2020.01.008>

Chen, C. W., & Sha, D. Y. (1999). A design approach to the multi-objective facility layout problem. *International Journal of Production Research*, 37(5), 1175-1196. Doi: <https://doi.org/10.1080/002075499191463>

Chen, C. W., & Sha, D. Y. (2005). Heuristic approach for solving the multi-objective facility layout problem. *International Journal of Production Research*, 43(21), 4493-4507. Doi: <https://doi.org/10.1080/00207540500056383>

Dong, X., Chen, P., Huang, H., & Nowak, M. (2013). A multi-restart iterated local search algorithm for the permutation flow shop problem minimizing total flow time. *Computers & Operations Research*, 40(2), 627-632. Doi: <https://doi.org/10.1016/j.cor.2012.08.021>

Dutta, K. N., & Sahu, S. (1982). A multigoal heuristic for facilities design problems: MUGHAL. *International Journal of Production Research*, 20(2), 147-154. Doi: <https://doi.org/10.1080/00207548208947756>

- Fortenberry, J. C., & Cox, J. F. (1985). Multiple criteria approach to the facilities layout problem. *International Journal of Production Research*, 23(4), 773-782. Doi: <https://doi.org/10.1080/00207548508904746>
- Harmonosky, C. M., & Tothoro, G. K. (1992). A multi-factor plant layout methodology, *International Journal of Production Research*, 30(8), 1773-1789. Doi: <https://doi.org/10.1080/00207549208948121>
- Hosseini-Nasab, H., Fereidouni, S., Ghomi, S. M. T. F., & Fakhrzad, M. B. (2018). Classification of facility layout problems: a review study. *The International Journal of Advanced Manufacturing Technology*, 94(1), 957-977. Erişim adresi: <https://link.springer.com/article/10.1007/s00170-017-0895-8>
- Khare, V. K., Khare, M. K., & Neema, M. L. (1988). Combined computer-aided approach for the facilities design problem and estimation of the distribution parameter in the case of multigoal optimization. *Computers & Industrial Engineering*, 14(4), 465-476. Doi: [https://doi.org/10.1016/0360-8352\(88\)90048-4](https://doi.org/10.1016/0360-8352(88)90048-4)
- Lourenço, H. R., Martin, O. C., & Stützle, T. (2019). Iterated Local Search: Framework and Applications. *Handbook of Metaheuristics*, 129-168. Erişim adresi: https://link.springer.com/chapter/10.1007/978-3-319-91086-4_5
- Malakooti, B., & Tsurushima, A. (1989). An expert system using priorities for solving multiple-criteria facility layout problems. *International Journal of Production Research*, 27(5), 793-808. Doi: <https://doi.org/10.1080/00207548908942588>
- Matai, R., Singh, S. P., & Mittal, M. L. (2013). Modified simulated annealing based approach for multi objective facility layout problem. *International Journal of Production Research*, 51(14), 4273-4288. Doi: <https://doi.org/10.1080/00207543.2013.765078>
- Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11), 1097-1100. Doi: [https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2)
- Michallet, J., Prins, C., Amodeo, L., Yalaoui, F., & Vitry, G. (2014). Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers & Operations research*, 41, 196-207. Doi: <https://doi.org/10.1016/j.cor.2013.07.025>
- Mohamadi, A., Ebrahimnejad, S., & Tavakkoli-Moghaddam, R. (2018). A novel two-stage approach for solving a bi-objective facility layout problem. *International Journal of Operational Research*, 31(1), 49-87. Doi: <https://doi.org/10.1504/IJOR.2018.088557>
- Penna, P. H. V., Subramanian, A., & Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2), 201-232. Erişim adresi: <https://link.springer.com/article/10.1007/s10732-011-9186-y>
- Rosenblatt, M. J. (1979). The facilities layout problem: a multi-goal approach. *International Journal of Production Research*, 17(4), 323-332. Doi: <https://doi.org/10.1080/00207547908919617>
- Sahni, S., & Gonzalez, T. (1976). P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3), 555-565. Doi: <https://doi.org/10.1145/321958.321975>
- Sha, D. Y., & Chen, C. W. (2001). A new approach to the multiple objective facility layout problem. *Integrated Manufacturing Systems*, 12(1), 59-66. Doi: <https://doi.org/10.1108/09576060110361546>
- Shojaie, A. A., & Bariran, S. E. S. (2020). Comparison of two modified meta-heuristic soft algorithms for solving a bi-objective facility layout problem. *International Journal of Mathematics in Operational Research*, 16(3), 435-454. Doi: <https://doi.org/10.1504/IJMOR.2020.106928>
- Singh, S. P., & Sharma, R. R. (2006). A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology*, 30(5-6), 425-433. Erişim adresi: <https://link.springer.com/article/10.1007/s00170-005-0087-9>
- Stützle, T. (2006). Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, 174(3), 1519-1539. Doi: <https://doi.org/10.1016/j.ejor.2005.01.066>
- Subramanian, A., Drummond, L. M. D. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers &*

Operations Research, 37(11), 1899-1911. Doi: <https://doi.org/10.1016/j.cor.2009.10.011>

Suresh, G., & Sahu, S. (1993). Multiobjective facility layout using simulated annealing. *International Journal of Production Economics*, 32(2), 239-254. Doi: [https://doi.org/10.1016/0925-5273\(93\)90071-R](https://doi.org/10.1016/0925-5273(93)90071-R)

Şahin, R. (2011). A simulated annealing algorithm for solving the bi-objective facility layout problem. *Expert Systems with Applications*, 38(4), 4460-4465. Doi: <https://doi.org/10.1016/j.eswa.2010.09.117>

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). *Facilities planning*. John Wiley & Sons. Erişim adresi: <https://books.google.com.tr/books?id=-xBIq6Qm2SQC&lpg=PA2&ots=sD0AicFLY5&dq=f>

[facilities%20planning%20tompkins&lr&hl=tr&pg=PA2#v=onepage&q=facilities%20planning%20tompkins&f=false](https://search.proquest.com/docview/199910723?pq-origsite=gscholar&fromopenview=true)

Urban, T. L. (1989). Combining qualitative and quantitative analyses in facility layout. *Production and Inventory Management*, 30(3), 73-77. Erişim adresi: <https://search.proquest.com/docview/199910723?pq-origsite=gscholar&fromopenview=true>

Zohali, H., Naderi, B., Mohammadi, M., & Roshanaei, V. (2019). Reformulation, linearization, and a hybrid iterated local search algorithm for economic lot-sizing and sequencing in hybrid flow shop problems. *Computers & Operations Research*, 104, 127-138. Doi: <https://doi.org/10.1016/j.cor.2018.12.008>

Ekler

Ek Tablo 1

Veri Kümesi 1 için 1. Tip Normalizasyon ile Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim							
1	0	2	4	7	1	6	3	5	8
0,9	0,1	6	3	1	8	2	4	5	7
0,8	0,2	8	5	4	2	7	1	3	6
0,7	0,3	6	3	4	2	1	7	5	8
0,6	0,4	6	3	5	2	1	7	4	8
0,5	0,5	1	7	4	5	6	3	2	8
0,4	0,6	5	4	7	1	8	2	3	6
0,3	0,7	8	2	3	6	5	4	7	1
0,2	0,8	1	7	4	5	6	3	2	8
0,1	0,9	6	3	2	8	1	7	4	5
0	1	8	2	3	6	5	4	7	1

Ek Tablo 2

Veri Kümesi 1 için 2. Tip Normalizasyon ile Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim							
1	0	2	4	7	1	6	3	5	8
0,9	0,1	8	1	3	6	7	5	4	2
0,8	0,2	8	5	4	2	7	1	3	6
0,7	0,3	6	3	4	2	1	7	5	8
0,6	0,4	7	3	4	2	1	6	5	8
0,5	0,5	8	2	3	6	5	4	7	1
0,4	0,6	8	2	3	6	5	4	7	1
0,3	0,7	5	4	7	1	8	2	3	6
0,2	0,8	5	4	7	1	8	2	3	6
0,1	0,9	1	7	4	5	6	3	2	8
0	1	5	4	7	1	8	2	3	6

Ek Tablo 3

Veri Kümesi 2 için 1. Tip Normalizasyon ile Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim											
1	0	8	1	10	6	2	11	3	7	9	12	4	5
0,9	0,1	6	10	1	5	7	3	11	8	2	4	12	9
0,8	0,2	6	7	9	5	10	2	1	12	4	3	11	8
0,7	0,3	4	3	11	8	2	10	1	12	7	6	9	5
0,6	0,4	5	9	6	7	12	1	10	2	8	11	3	4
0,5	0,5	7	6	9	5	4	2	1	12	3	11	8	10
0,4	0,6	7	6	9	5	4	2	1	12	3	11	8	10
0,3	0,7	10	8	11	3	12	1	2	4	5	9	6	7
0,2	0,8	10	11	3	4	1	8	6	2	12	9	5	7
0,1	0,9	4	3	11	10	2	6	8	1	7	5	9	12
0	1	3	11	8	10	4	6	9	1	7	2	5	12

Ek Tablo 4

Veri Kümesi 2 için 2. Tip Normalizasyon ile Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim											
1	0	9	12	4	2	8	1	11	6	5	3	10	7
0,9	0,1	8	12	11	4	9	1	3	10	5	2	7	6
0,8	0,2	4	3	11	8	2	10	1	12	7	6	9	5
0,7	0,3	8	11	3	4	12	1	10	2	5	9	6	7
0,6	0,4	5	9	6	7	12	1	10	2	8	11	3	4
0,5	0,5	7	6	9	5	4	2	1	12	3	11	8	10
0,4	0,6	4	3	11	8	7	6	9	5	2	10	1	12
0,3	0,7	3	11	8	10	4	2	1	12	7	6	9	5
0,2	0,8	4	3	11	10	2	6	8	1	7	5	9	12
0,1	0,9	7	5	9	12	2	6	8	1	4	3	11	10
0	1	4	7	9	12	6	2	5	1	3	11	8	10

Ek Tablo 5

Veri Kümesi 3 için 1. Tip Normalizasyon ile Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim														
1	0	12	7	14	15	6	13	9	1	8	11	10	5	3	2	4
0,9	0,1	10	5	3	2	4	13	9	1	8	11	12	7	14	15	6
0,8	0,2	10	4	13	9	12	3	2	8	15	7	5	11	1	14	6
0,7	0,3	3	10	2	4	15	5	13	8	9	7	11	1	14	6	12
0,6	0,4	3	11	1	14	12	5	8	2	6	7	10	13	4	9	15
0,5	0,5	12	7	6	14	5	15	10	2	8	3	9	4	13	1	11
0,4	0,6	12	7	6	14	5	15	10	2	8	3	9	4	13	1	11
0,3	0,7	11	13	1	4	9	3	8	2	10	15	5	14	6	7	12
0,2	0,8	4	1	13	11	3	9	10	2	8	5	15	7	6	12	14
0,1	0,9	15	7	6	12	14	9	10	2	8	5	4	1	13	11	3
0	1	15	7	6	12	14	9	10	2	8	5	4	1	13	11	3

Ek Tablo 6

Veri Kümesi 3 için 2. Tip Normalizasyon ile Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim														
1	0	12	7	14	15	6	13	9	1	8	11	10	5	3	2	4
0,9	0,1	6	15	14	7	12	11	8	1	9	13	4	2	3	5	10
0,8	0,2	3	10	2	4	15	5	13	8	9	7	11	1	14	6	12
0,7	0,3	11	1	14	6	12	5	13	8	9	7	3	10	2	4	15
0,6	0,4	12	14	1	11	3	7	6	2	8	5	15	9	4	13	10
0,5	0,5	11	1	13	4	9	3	8	2	10	15	5	14	6	7	12
0,4	0,6	9	4	13	1	11	15	10	2	8	3	12	7	6	14	5
0,3	0,7	5	14	6	7	12	3	8	2	10	15	11	13	1	4	9
0,2	0,8	3	11	13	1	4	5	8	2	10	9	14	12	6	7	15
0,1	0,9	3	11	13	1	4	5	8	2	10	9	14	12	6	7	15
0	1	15	7	6	12	14	9	10	2	8	5	4	1	13	11	3

Ek Tablo 7

Veri Kümesi 4 için 1. Tip Normalizasyon ile Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim																			
1	0	9	14	10	3	12	7	2	1	4	19	13	17	5	8	15	20	16	18	11	6
0,9	0,1	16	13	7	8	9	11	17	2	3	15	20	14	1	4	6	18	5	10	12	19
0,8	0,2	18	5	10	12	19	20	14	1	4	6	11	17	2	3	15	16	13	7	9	8
0,7	0,3	20	4	1	7	8	6	11	10	9	16	12	18	14	2	3	15	5	19	17	13
0,6	0,4	15	13	3	7	8	17	9	12	16	6	19	2	14	11	20	5	18	10	1	4
0,5	0,5	5	18	10	1	4	19	2	14	11	20	17	9	12	16	6	15	13	3	7	8
0,4	0,6	5	10	1	18	4	19	2	14	11	20	17	9	16	12	6	15	7	3	13	8
0,3	0,7	15	7	3	13	8	17	9	16	12	6	19	2	14	11	20	5	10	1	18	4
0,2	0,8	4	18	1	10	5	20	11	14	2	19	6	12	16	9	17	8	13	3	7	15
0,1	0,9	4	11	18	1	5	20	13	14	2	10	6	16	12	9	19	8	3	17	7	15
0	1	5	15	12	6	4	1	18	14	11	20	10	2	9	16	13	19	17	7	3	8

Ek Tablo 8

Veri Kümesi 4 için 2. Tip Normalizasyon ile Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim																			
1	0	12	15	19	5	18	3	8	2	17	13	14	10	1	7	16	9	6	4	11	20
0,9	0,1	18	5	10	12	19	20	14	1	4	6	11	17	2	3	15	16	13	7	9	8
0,8	0,2	6	8	9	15	13	11	7	2	3	17	4	1	10	14	16	20	18	5	12	19
0,7	0,3	13	17	19	15	5	3	2	14	12	18	16	9	10	6	11	8	7	1	4	20
0,6	0,4	20	4	12	18	5	11	1	10	14	19	6	16	9	2	15	8	7	3	13	17
0,5	0,5	17	7	3	13	8	15	2	9	16	6	19	10	14	12	11	5	1	18	4	20
0,4	0,6	4	18	1	10	5	20	11	14	2	19	6	12	16	9	17	8	13	3	7	15
0,3	0,7	15	7	3	13	8	17	9	16	12	6	19	2	14	11	20	5	10	1	18	4
0,2	0,8	4	18	1	10	5	20	11	14	2	19	6	12	16	9	17	8	13	3	7	15
0,1	0,9	5	1	18	11	4	10	2	14	13	20	19	9	12	16	6	15	7	17	3	8
0	1	10	17	7	3	8	19	2	9	16	13	1	18	14	11	20	5	15	12	6	4

Ek Tablo 9

Veri Kümesi 5 için Üretilen Çözümler

w_1	w_2	YYA ile bulunan yerleşim								w_1	w_2	YYA ile bulunan yerleşim									
1	0	33	37	28	35	11	31	3	36	0,4	0,6	37	3	35	16	29	14	36	9		
		22	10	25	17	2	20	40	1			33	34	18	24	10	28	12	27		
		30	18	16	29	15	13	9	21			22	6	23	1	15	17	4	40		
		14	12	34	39	32	23	8	5			32	20	25	13	19	5	8	38		
		7	19	27	24	6	38	4	26			30	11	31	2	21	7	39	26		
0,9	0,1	11	6	39	35	20	21	4	26	0,3	0,7	22	32	37	33	35	28	29	9		
		7	29	15	2	9	31	8	38			30	3	34	23	1	24	16	14		
		22	37	3	17	25	40	5	1			31	20	2	18	17	10	12	27		
		30	18	34	27	16	13	23	36			11	7	8	25	15	5	4	40		
		33	14	12	10	28	32	19	24			26	21	39	13	6	19	36	38		
0,8	0,2	22	7	15	9	8	40	4	26	0,2	0,8	9	29	28	24	33	37	32	22		
		11	6	39	3	20	31	38	21			27	16	10	18	23	34	3	20		
		37	29	2	17	35	25	5	1			14	12	35	1	25	2	31	30		
		30	18	34	16	23	13	28	36			40	4	5	17	15	8	7	11		
		33	14	12	27	10	32	24	19			38	36	19	6	13	39	21	26		
0,7	0,3	26	38	5	1	13	36	21	30	0,1	0,9	9	29	28	24	33	37	32	22		
		4	8	19	29	15	25	6	11			27	16	10	18	23	34	3	20		
		40	7	23	16	39	31	20	22			14	12	35	1	25	2	31	30		
		28	12	24	10	18	34	32	33			40	4	5	15	6	13	7	11		
		9	27	14	17	2	35	3	37			38	36	19	17	8	39	21	26		
0,6	0,4	21	26	7	39	6	20	11	22	0	1	9	29	24	23	15	3	7	22		
		38	4	8	19	15	25	31	30			16	4	10	1	25	18	11	31		
		1	5	40	17	2	13	23	32			14	12	5	33	6	34	20	30		
		36	29	12	24	16	18	34	33			40	36	28	35	19	8	2	32		
		14	9	28	27	35	10	3	37			27	38	37	17	13	21	39	26		
0,5	0,5	22	11	20	39	7	21	8	26												
		30	31	6	13	15	19	4	38												
		32	23	25	2	1	36	5	40												
		33	34	18	16	24	29	12	27												
		37	3	35	10	17	14	28	9												