Endüstri Mühendisliği Dergisi
Cilt: 28 Sayı: 1 Sayfa: (19-39) Makale

ÇOK AMAÇLI TAMSAYI PROGRAMLAMA PROBLEMLERİ İÇİN TEMSİLİ ÇÖZÜM ÜRETEN YAKLAŞIMLARIN VE KALİTE ÖLÇÜLERİNİN İNCELENMESİ

Banu LOKMAN

Orta Doğu Teknik Üniversitesi, Endüstri Mühendisliği Bölümü, Ankara
lbanu@metu.edu.tr

Geliş Tarihi: 02.11.2016; Kabul Ediliş Tarihi: 05.05.2017

ÖZ

Çok amaçlı tamsayı programlama problemlerinde baskın noktaların sayısı problemin büyüklüğüne bağlı olarak üssel bir büyüme gösterir. Bu nedenle, bu problemler için tüm baskın noktaları bulmak zordur ve karar verici için pratik bir yaklaşım da değildir. Tüm baskın noktalar yerine, bu noktaları belirli kalite ölçülerine göre iyi temsil eden noktalar bulmak önemlidir. Bu çalışmamızda, temsili kümenin değerlendirilmesinde kullanılan kalite ölçülerini ve bu kalite ölçülerine göre tüm baskın nokta kümesini iyi temsil eden noktalar bulan yaklaşımları inceleyeceğiz.

Anahtar Kelimeler: Baskın nokta, temsili baskın nokta, kalite ölçüleri, çok amaçlı tamsayı programlama, çok amaçlı bileşi optimizasyonu

A SURVEY ON FINDING REPRESENTATIVE POINTS FOR MULTI-OBJECTIVE INTEGER PROGRAMS AND QUALITY MEASURES

ABSTRACT

The number of nondominated points of multi-objective integer programming problems increases exponentially with the problem size. Therefore, finding all nondominated points is computationally hard and not practical for the decision maker. Instead of generating all nondominated points, it is reasonable to generate a set of points that represents the nondominated set with a desired quality level. In this study, we review the quality measures used to evaluate the representative sets and the approaches that generate representative points.

Keywords: Nondominated point, representative point, quality measures, multi-objective integer programming, multi-objective combinatorial optimization

1. GİRİŞ

Günümüzde birçok organizasyonda, karar vericilerin (KV) karşılaştıkları optimizasyon problemleri, doğası gereği çoğu zaman birden fazla ve birbiri ile çelişen amaçlar içerir. Yatırım planlama, yer seçimi, çizelgeleme, rotalama, ulaştırma gibi birçok alanda uygulaması olan çok amaçlı tamsayı programlama (ÇATP) problemleri, çelişen amaçlar nedeniyle, tek bir optimal nokta yerine *baskın nokta* olarak tanımlanan çok sayıda çözüm içerir (Özçelik ve Saraç, 2011; Aktaş vd., 2011; Koçanlı vd., 2012; Kamışlı Öztürk vd., 2016). Baskın noktalar, en az bir amaç fonksiyonundan ödün vermeden başka bir amaç fonksiyonunda iyileştirme yapılması mümkün olmayan noktalardır.

ÇATP problemlerinde problem büyüklüğü arttıkça, herhangi bir baskın çözümü bulmak zorlaştığı gibi aynı zamanda baskın nokta sayısı da üssel bir büyüme gösterir (Ehrgott ve Gandibleux, 2000). Bu nedenle, ÇATP problemleri için birçok sezgisel, metasezgisel yaklaşım ve yakınsama algoritmaları geliştirilmiştir. Ehrgott ve Gandibleux (2004), özel ÇATP problemleri olan Çok Amaçlı Bileşi Optimizasyonu (ÇABO) problemleri için özel geliştirilmiş bu algoritmaları incelerken, Ruzika ve Wiecek (2005) ise çok amaçlı optimizasyon (ÇAO) problemleri üzerine bir inceleme sunmaktadır.

Son dönemde, tüm baskın noktaları bulan etkili algoritmalar geliştirilmiştir; ancak tüm baskın noktaları bulmak ve KV'ye sunmak hem zor hem de pratik olmayan bir yaklaşımdır (Lokman ve Köksalan, 2013; Mavrotas ve Florios, 2013; Kırlık ve Sayın, 2014; Özlen vd. 2014; Dächert ve Klamroth, 2015). Bu nedenle, tüm baskın noktaları bulmak yerine, belirli kalite ölçülerini baz alarak tüm baskın nokta kümesini iyi temsil eden bir alt küme bulmak ÇATP problemleri için önemlidir.

Bu çalışmamızda, ÇATP ve ÇABO problemleri için KV tarafından belirlenecek kalitede bir temsili baskın nokta kümesi bulmayı hedefleyen yaklaşımlar ve önerilen kalite ölçüleri incelenecektir. İlk olarak, 2. kısımda gerekli tanımlar verilerek temel yaklaşımlar sunulacaktır. 3. kısımda temsili kümelerin değerlendirilmesinde kullanılan kalite ölçüleri ve 4. kısımda, geliştirilen yaklaşımlar incelenecektir. 5. kısımda ise sonuçlar sunulacaktır.

2. TANIMLAR VE GENEL BİLGİLER

Genel p amaçlı bir tamsayı optimizasyon problemi aşağıdaki şekilde tanımlanabilir:

(ÇATP)

"Maks" $z = \left(z_1(\mathrm{x}), \ldots, z_p(\mathrm{x})\right)$

Kısıtlar

$\mathrm{x} \in \mathrm{X}$

Burada x, karar vektörünü; $\mathrm{X} \in \mathbb{Z}^n$, tamsayı olurlu karar vektörü kümesini; $z_j(x)$ o, karar vektörü için j. amaç fonksiyonunun aldığı değeri ve $z(x) = (z_1(x), \ldots, z_p(x))$, x çözümüne karşılık gelen amaç vektörünü temsil etmektedir. X kümesinin amaç fonksiyonu uzayındaki görüntüsü ise Z kümesi ile tanımlanmaktadır. Tırnak işaretleri vektör maksimizasyonunun tanımlı bir matematiksel operasyon olmadığını ifade etmektedir.

Tanım 1: Herhangi $x_1, x_2 \in$ X çözümü için, $z_j(x_1) \leq z_j(x_2)$ $j = 1, \ldots, p$ en az bir amaç fonksiyonunda $z_j(x_1) < z_j(x_2)$ koşulları sağlanıyorsa, x_2 çözümünün x_1 çözümünü *baskıladığı* söylenir. Böyle bir x_2 çözümü yoksa, x_1 çözümüne *etkin çözüm* ve $z(x_1)$ noktasına da *baskın nokta* denir.

Tanım 2: Herhangi $x_1, x_2 \in \mathbf{X}$ çözümü için, $z_j(x_1) < z_j(x_2)$ $j = 1, \ldots, p$ koşulu sağlanıyorsa, x_2 çözümünün x_1 çözümünü *tam baskıladığı* söylenir. Eğer böyle bir x_2 çözümü yoksa, x_1 çözümüne *zayıf etkin çözüm* ve $z(x_1)$ noktasına *zayıf baskın nokta* denir.

Tüm baskın noktalar aynı zamanda zayıf baskın noktalardır.

Tüm baskın noktaları veya bir kısmını bulmak için önerilen birçok yöntem vardır. Bu yöntemlerden biri, amaç fonksiyonlarını pozitif ağırlıklandırarak çok amaçlı optimizasyon problemini tek amaçlı optimizasyon problemine dönüştürmektir. Bu şekilde, farklı pozitif ağırlıklar kullanarak bulunabilen noktalara, *destekli baskın nokta* denir. Öyle ki, her destekli baskın nokta, $\mathrm{z^w}$, için $\mathrm{w}^T\mathrm{z^w} = \mathrm{maks}_{\mathrm{x}\in\mathrm{X}} \sum_{k=1}^{p} w_k z_k(\mathrm{x})$ eşitliğini sağlayan pozitif bir ağırlık vektörü w > 0 bulunabilir. Ancak, böyle bir ağırlık vektörü bulunamayan, bu nedenle *ağırlıklandırma yöntemi* ile bulunması mümkün olmayan noktalar, *desteksiz baskın noktalar* da vardır. Destekli noktaların elde edilmesinden sonra, olası bölgelerde desteksiz noktaların aranması için problem tiplerine özel yöntemler geliştirilmiştir (Ulungu ve Teghem, 1995; Visee vd., 1998; Steiner ve Radzik, 2008; Sipahioğlu ve Saraç, 2010). Ancak, bu yöntemler genel ÇATP problemleri için uygulanabilir değildir.

Baskın noktaların bulunması için uygulanan bir diğer yöntem *epsilon-kısıtı yöntemi*dir. Bu yöntemde, bir kriter amaç fonksiyonu olarak seçilirken diğer kriterler için alt sınır koyan kısıtlar eklenmektedir:

$(P^{\varepsilon j})$

Maks $z_j(x)$

Kısıtlar

$$z_k(\mathrm{x}) \geq \varepsilon_k \quad k = 1, \dots, p \quad k \neq j$$

$$\mathrm{x} \in \mathrm{X}$$

Ancak, amaç fonksiyonunda diğer kriterlerin bulunmaması nedeniyle, bu yöntem ile baskın olmayan zayıf baskın noktalar da bulunabilir. Bunu engellemek amacıyla *modifiye edilmiş epsilon-kısıtı* yöntemi kullanılır:

$(P'^{\varepsilon j})$

$$\text{Maks } z_j(\mathbf{x}) + \sum_{\substack{k=1 \\ k \neq j}}^{p} \rho_k z_k(\mathbf{x})$$

Kısıtlar

$$z_k(\mathrm{x}) \geq \varepsilon_k \quad k = 1, \dots, p \quad k \neq j$$

$$\mathrm{x} \in \mathrm{X}$$

Burada $\rho_k > 0$, baskın olmayan zayıf baskın noktaların bulunmasını engellemek amacıyla yeterince küçük pozitif bir sabit olarak seçilir (Steuer, 1986: 429-430).

$X_E \in X$ ve $Z_B \in Z$ sırasıyla tüm etkin çözümlerin ve baskın noktaların kümesini temsil etmektedir. Özellikle ÇATP problemlerinde, farklı etkin çözümlerin kriter uzayında aynı baskın noktaya karşılık gelmesi mümkündür. Bu nedenle, bu iki kümenin eleman sayısı arasında $|X_E| \geq |Z_B|$ ilişkisi vardır. Farklı amaç fonksiyonlarının birlikte değerlendirilmesi ve karşılaştırılması için her amaç fonksiyonunun baskın noktalar içerisinde alabileceği en iyi ve en kötü değerleri bulmak önemlidir.

Tanım 3: Her amaç fonksiyonunun alabileceği en iyi değerlerden oluşan noktaya *ideal nokta* denir. İdeal nokta, $\mathrm{z}^I = (z_1^I, \dots, z_p^I)$ ile gösterilir ve $z_j^I = \text{maks}_{\mathrm{x} \in \mathrm{X}}\, z_j(\mathrm{x})\ j = 1, \dots, p$ 'dir.

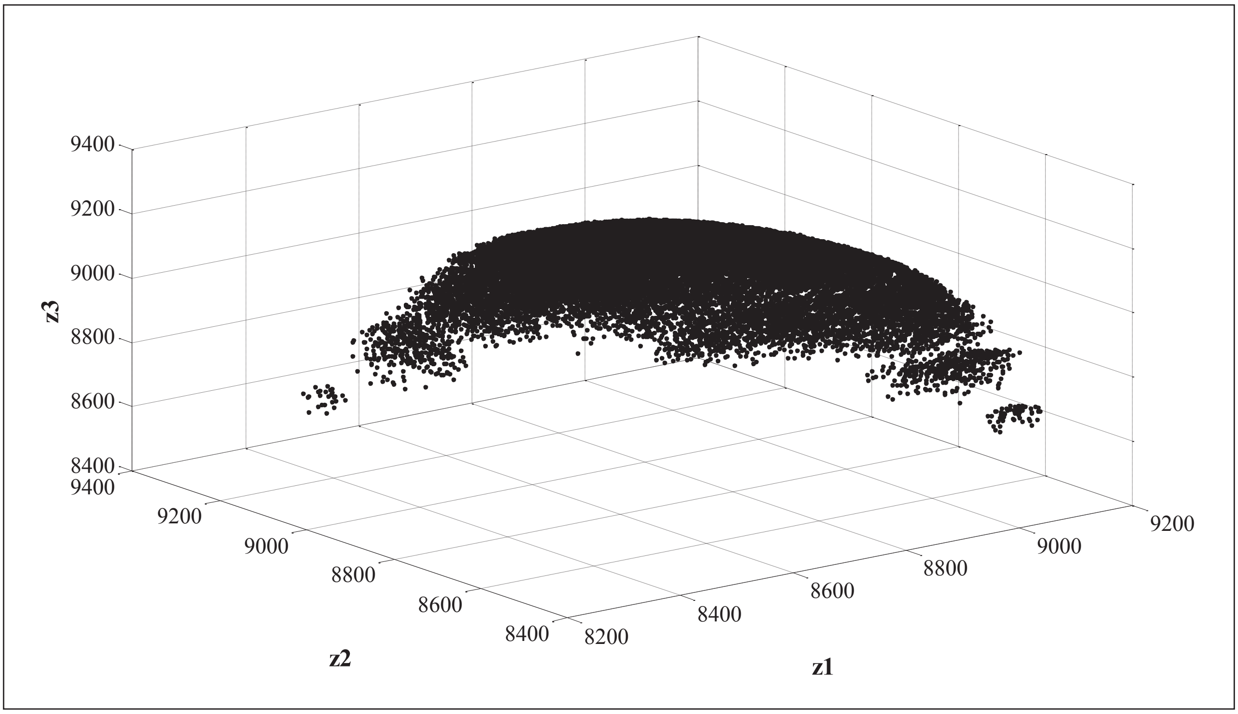
Tanım 4: Her amaç fonksiyonunun etkin çözümler içerisinde alabileceği en kötü değerlerden oluşan noktaya *nadir nokta* denir. Nadir nokta, $\mathrm{z}^N = (z_1^N, \dots, z_p^N)$ ile gösterilir ve $z_j^N = \min_{\mathrm{x} \in \mathrm{X}_E} z_j(\mathrm{x})\ j = 1, \dots, p$'dir.

İdeal nokta, her bir amaç fonksiyonunun sırasıyla en iyilendiği tek amaçlı problemler çözülerek bulunabilir. Ancak, etkin çözümler üzerinde en kötü değerlere karşılık gelen nadir noktayı bulmak zordur. Son dönemde bunun için geliştirilmiş özel algoritmalar bulunmaktadır (Jorge, 2009; Köksalan ve Lokman, 2015; Kırlık ve Sayın, 2015). İdeal ve nadir noktalara, amaç fonksiyonlarının ölçeklendirilmesinde kullanılmaları nedeniyle, ÇATP problemleri için tasarlanmış birçok algoritmada ihtiyaç duyulur (Masin ve Bukchin, 2008; Karasakal ve Köksalan, 2009; Miettinen vd., 2010).

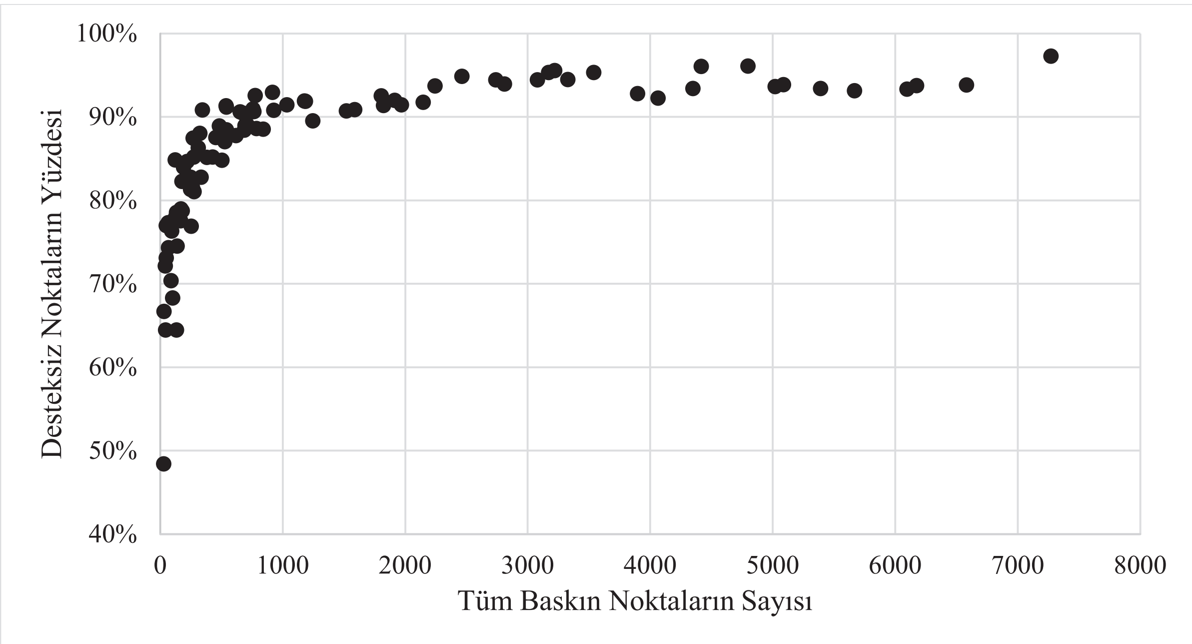
3. KALİTE ÖLÇÜLERİNİN İNCELENMESİ

Tüm baskın noktaların sayısı problem büyüklüğüne, amaç fonksiyonlarının sayısına bağlı olarak hızla artmaktadır. Şekil 1'de de gösterildiği gibi, örneğin 200 parçalı, 3 amaçlı bir sırt çantası problemi için 27.260 baskın nokta vardır. Ayrıca, Şekil 2'de gösterildiği gibi, problem büyüklüğü arttıkça, ağırlıklandırma yöntemi ile bulunamayan desteksiz baskın noktaların toplam baskın nokta sayısına oranı da giderek artmaktadır. Bu nedenle, özellikle giderek daha karmaşık hale gelen günümüz problemlerini de düşünerek tüm baskın noktaları bulmak yerine temsili bir nokta kümesi, $Z_T \in \mathbb{R}^p$ veya temsili çözüm kümesi, $X_T \in \mathbb{R}^n$ bulmak pratik ve uygulanabilir bir yaklaşımdır. Temsili küme, baskın noktalardan, baskılanan olurlu noktalardan veya olurlu noktalara yakın yaklaşık noktalardan oluşabilir.

Temsili kümenin kalitesinin değerlendirilmesinde farklı performans ölçüleri tanımlanmıştır (Sayın, 2000; Wu ve Azarm, 2001; Zitzler vd., 2003). Sayın (2000), temsili kümenin kalitesini değerlendirmek için temsili kümenin eleman sayısı, kapsama özelliği ve dağılımı olarak temel üç farklı ölçü önermiştir. Eleman sayısı ölçüleri, her yeni bir nokta bulmanın getirdiği çözüm



Şekil 1. 200 Parçalı Üç Amaçlı Bir Sırt Çantası Problemine ait Baskın Çözümler (|Z_B|=27.260)



Şekil 2. Desteksiz Baskın Noktaların Oranı*

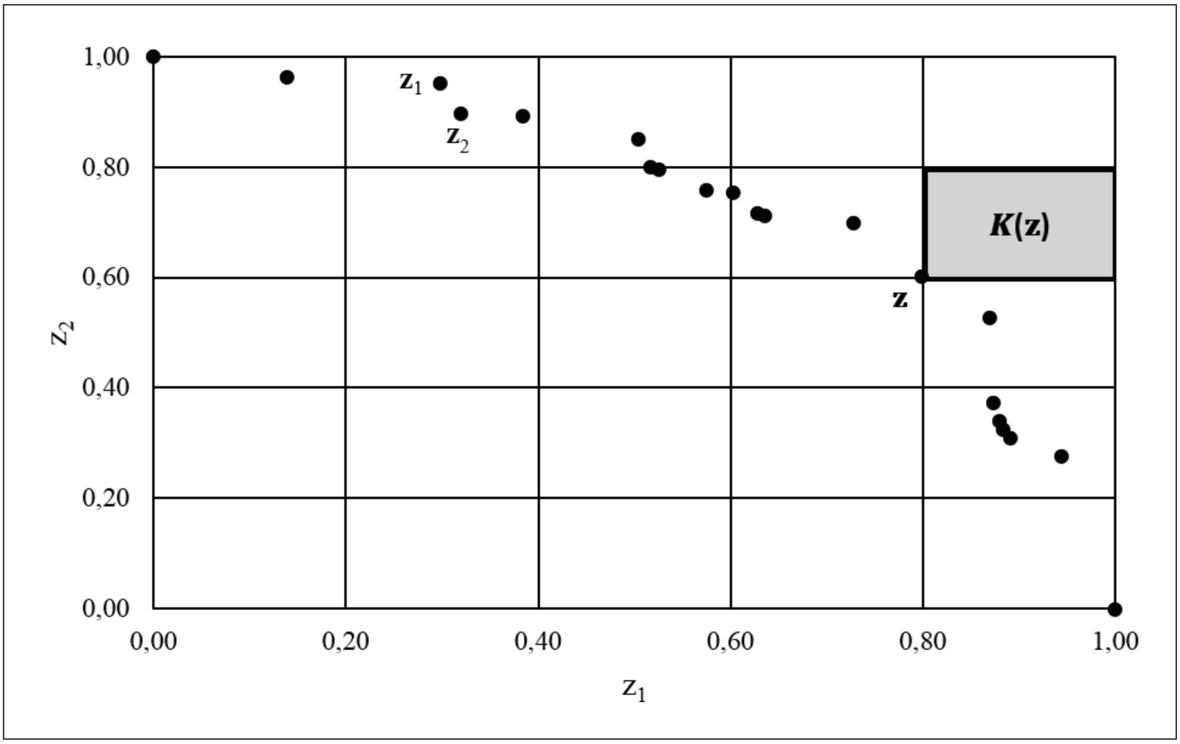
*Toplam 90 Problem - 10'ar üç amaçlı atama problemi (10, 20, 30-görevli), sırt çantası problemi (25, 50, 100-parçalı), en kısa yol problemi (50, 100, 200-düğümlü) zorluğu nedeniyle temsili kümenin büyüklüğünü ölçmektedir. Her bir baskın noktanın ne kadar iyi temsil edildiği kapsama ölçüleri ile, üretilen temsili noktaların amaç fonksiyonu uzayına dağılımı ise dağılım ölçüleri ile değerlendirilmektedir. Verilen temsili bir küme için, eleman sayısı ve dağılım ölçüleri hesaplanabilirken, kapsama ölçülerinin bulunabilmesi için baskın nokta kümesinin veya baskın noktalara yakın olduğu bilinen başka bir kümenin bilinmesi gerekmektedir.

Faulkenberg ve Wiecek (2010), genel çok amaçlı optimizasyon problemleri için kullanılabilecek kalite ölçülerini Sayın'ın (2000) önerdiği gibi üç farklı grupta değerlendirmişlerdir. Bu çalışmalara benzer olarak, burada da kalite ölçüleri eleman sayısı, kapsama ve dağılım ölçüleri olarak üç grupta sınıflandırılarak incelenecektir.

3.1 Eleman Sayısı Ölçüleri

Temsili alt kümenin eleman sayısını ölçmek için üretilen nokta sayısı, $|Z_T|$ veya üretilen çözüm sayısı, $|X_T|$ olarak iki farklı ölçü kullanılabilir. Tek bir çözüm bulmanın bile zor olabildiği ÇATP problemlerinde, çoğu zaman tüm baskın nokta kümesini olabildiği kadar az nokta ile temsil etmek, $|Z_T| < |Z_B|$ veya $|X_T| < |X_E|$ hedeflenmektedir.

Bu ölçülerden farklı olarak, Wu ve Azarm (2001), yeni bir eleman sayısı ölçüsü tanımlamışlardır. Bu eleman sayısı ölçüsü, karar vericinin tercihlerine göre 0 ile 1 arasında seçilen bir μ değerine göre hesaplanır. Öyle ki, her bir amaç fonksiyonu (0, 1) arasında olacak şekilde normalize edilerek eşit 1/μ bölgeye ayrılmakta ve dolayısıyla amaç fonksiyonu uzayı toplam $1/\mu^p$ hiperküpe bölünmektedir. Aynı hiperküp içindeki çözümler arasındaki fark önemsiz olarak değerlendirilmekte ve tek nokta olarak sayılmaktadır. Şekil 3'te de gösterildiği gibi, aynı hiperküp içindeki çözümler, örneğin z_1 ve z_2 arasındaki fark önemsiz olarak değerlendirilmekte ve tek nokta olarak sayılmaktadır. Benzer şekilde, her z noktası için farksız değerlendirilen bölge, $K(z)$ belirlenerek değerlendirme yapılmakta ve bulunan farklı nokta sayısı temsili kümenin eleman sayısını oluşturmaktadır. Şekil 3'te verilen örnek için bu şekilde bulunan 9 farklı baskın nokta vardır.



Şekil 3. μ = 0,2 için z Noktasıyla Farksız Değerlendirilen Bölgeler, $K(z)$

3.2 Kapsama Ölçüleri

Kapsama ölçüleri, her bir baskın noktanın temsili alt küme tarafından ne kadar iyi temsil edildiğini değerlendirmek için kullanılan ölçülerdir. Bu nedenle, ilk olarak her bir baskın nokta, $z \in Z_B$ için temsili kümede yer alan en yakın nokta, $y \in Z_T$ temsilci olarak seçilmekte ve aralarındaki uzaklık, $u(z, y)$ dikkate alınmaktadır. Sayın (2000) tarafından önerilen ve $u(z, y)=\max_{i=1,\dots,p}(z_i-y_i)$ uzaklık ölçüsünün kullanıldığı kapsama ölçüsü Tanım 5'te verilmiştir.

Tanım 5: Herhangi bir temsili alt küme $Z_T \in \mathbb{R}^p$ içinde, tüm $z \in Z_B$ baskın noktalar için en fazla $\alpha \geq 0$ uzaklığında en az bir temsilci $y \in Z_T$ varsa, $u(z,y) \leq \alpha$, Z_T alt kümesine Z_B'nin *α-temsili* denir. Z_T alt kümesinin Z_B kümesini *kapsama hatası* $\alpha(Z_T) = \text{maks}_{z \in Z_B} \min_{y \in Z_T} u(z, y)$ şeklinde tanımlanır.

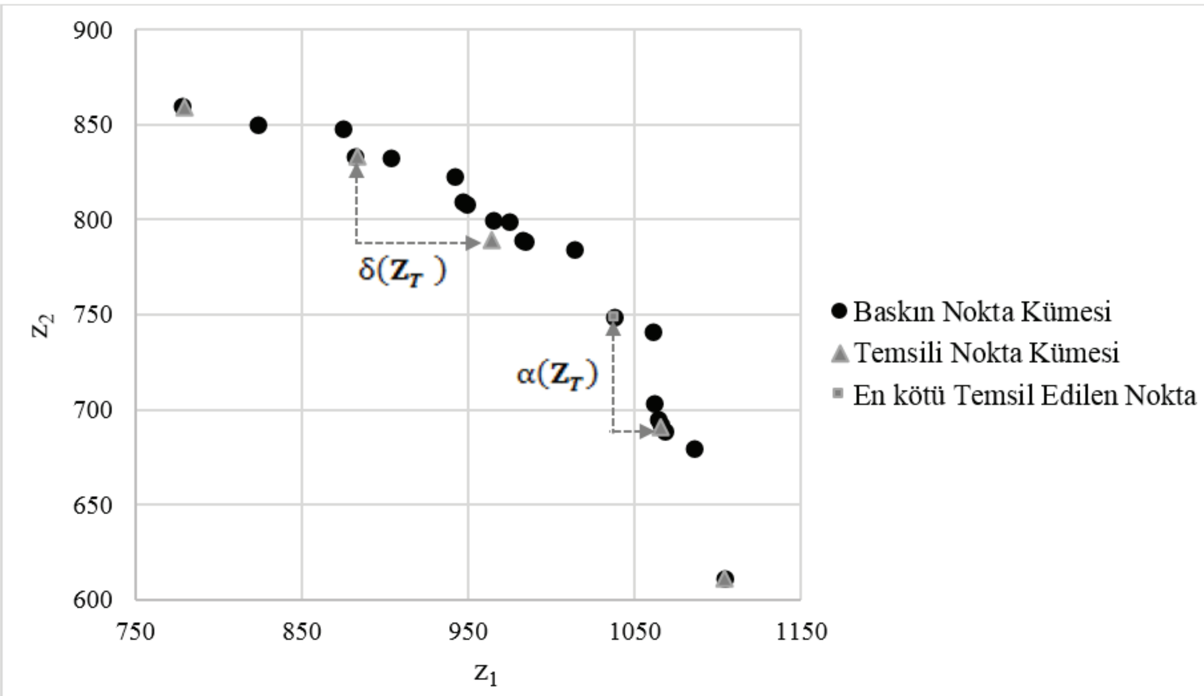
Sayın (2000), Şekil 4'te de gösterildiği gibi, kapsama ölçüsü için en kötü temsil edilen nokta üzerinden değerlendirme yaparken, Czyzżak ve Jaszkiweicz (1998) ise *ortalama temsil hatasını* hesaplamışlardır: $\frac{1}{|Z_B|}\sum_{z \in Z_B} \min_{y \in Z_T} u(z, y)$

Farklı olarak, Zitzler ve Thiele (1998), Z_T ve Z_B tarafından baskılanan bölgelerin toplam hacmini ölçerek, Tanım 6'da verilen kapsama ölçüsünü önermişlerdir. Hacim hesaplanırken, kriter uzayının tanımlanması için nadir noktası veya nadir noktasına yakın başka bir nokta referans alınmaktadır. Şekil 5, nadir noktası referans alınarak, Z_T ve Z_B tarafından baskılanan bölgeyi iki amaçlı bir problem üzerinde göstermektedir.

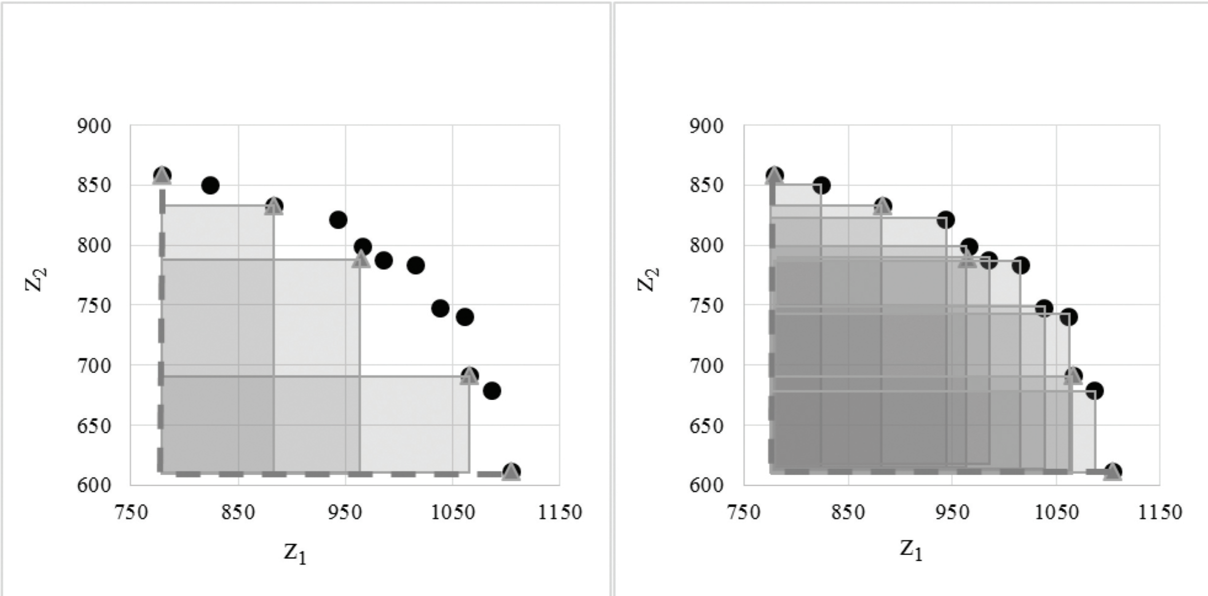
Tanım 6: Herhangi bir temsili alt küme $Z_T \in \mathbb{R}^p$ için üst-hacim göstergesi, Z_T tarafından baskılanan kriter uzayının hacminin, Z_B tarafından baskılanan kriter uzayının hacmine oranı $H(Z_T) = \frac{h(Z_T)}{h(Z_B)}$ olarak tanımlanır.

Benzer olarak, Boland vd. (2015) ve Boland vd. (2016), nadir noktasını referans alarak *üst hacim boşluğu*, $\frac{h(Z_B) - h(Z_T)}{h(Z_B)}$ değerini performans ölçüsü olarak kullanmışlardır.

Zitzler vd. (2003) tarafından önerilen bir diğer kapsama ölçüsü Tanım 7'de verilmiştir.



Şekil 4. Kapsama Hatasının ve Dağılım Derecesinin Hesaplanması



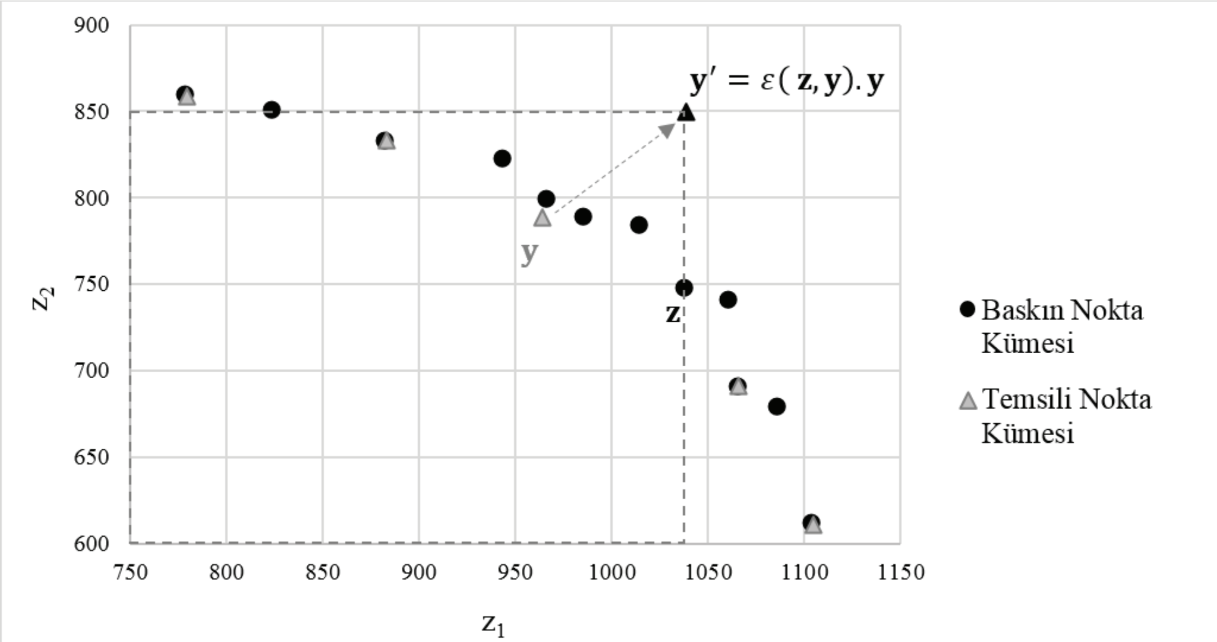
Şekil 5. Temsili Küme (Z_T) ve Baskın Küme (Z_B) Tarafından Baskılanan Bölgeler

Tanım 7: Herhangi bir temsili alt küme $Z_T \in \mathbb{R}^p$ için, tüm $z \in Z_B$ baskın noktaların baskılanmasını sağlamak için kullanılacak en küçük çarpan *epsilon göstergesi* olarak tanımlanır: $I_\varepsilon(Z_T) = \text{maks}_{z\in Z_B} \min_{y\in Z_T} \varepsilon(z, y)$ $= \text{maks}_{z\in Z_B} \min_{y\in Z_T} \underset{k=1,..,p}{\text{maks}} \left(\frac{z_k}{y_k}\right)$

Tanım 7'de de verildiği gibi, ε (*z, y*), baskın nokta (*z*) ve temsilci (*y*) için, $z \leq \varepsilon\ (z, y).y$ eşitsizliğini sağlayan en küçük çarpan olarak tanımlanmıştır (Şekil 6). Tüm baskın noktaların yer aldığı bir temsili küme, $Z_T=Z_B$ için $I_\varepsilon\ (Z_T)=1$ değerini almaktadır. ε (*z*, *y*), yakınsama algoritmalarında (1+ε) olarak kullanılan yakınsama oranına karşılık gelmektedir (Papadimitriou ve Yannakakis, 2000), ε-baskın küme, Z_ε olarak tanımlanan bu kümede her baskın nokta *z* için, $z \leq (1+\varepsilon)\ y$ koşulunu sağlayan bir $y \in Z_\varepsilon$ noktası vardır. Papadimitriou ve Yannakakis (2000), böyle bir temsili kümenin problem büyüklüğüne ve değerine göre polinom zamanda bulunabilmesi için yeterli ve gerekli koşulları tanımlamışlardır. Bu kalite ölçüsünün kullanıldığı temsili küme problemlerinde amaç, istenilen çözüm sayısı verildiğinde en küçük epsilon göstergesine sahip bir alt küme bulmak veya istenilen epsilon göstergesi verildiğinde eleman sayısı en küçük temsili küme bulmaktır. Kapsama hatası ile epsilon göstergesi birbiriyle yakından ilişkili kapsama ölçüleridir. Öyle ki, iyi kapsama ölçüsüne sahip bir temsili küme için iyi bir epsilon göstergesi değeri bulunabilir.

Zitzler (1999) ise iki nokta kümesinin birbirine göre dağılımını ölçen bir *kapsama göstergesi* önermiştir: $C(Z_1, Z_2) = \frac{|\{y_2 \in Z_2 | \exists y_1 \in Z_1 : y_1 \geq y_2\}|}{|Z_2|}$. $C(Z_1, Z_2)$, Z_2 kümesi içinde Z_1 kümesindeki noktalar tarafından "kapsanan", yani baskılanan veya eşit olan, noktaların oranını hesaplamaktadır. Bu kapsama ölçüsünü, baskın noktaları yaklaşık olarak bulan sezgisel/metasezgisel yöntemlerin kapsama performansını değerlendirmek için de kullanmak mümkündür (Deb, 2001: 311-312). $C\ (Z_T, Z_B)=1$, tüm baskın noktaların bulunması durumunda elde edilir.

Sezgisel veya metasezgisel yöntemlerde üretilen temsili küme, baskın veya baskılanan noktalardan oluşabilir ve bu nedenle temsili kümenin değerlendirilmesinde bulunan noktaların baskın noktalara olan uzaklığı ölçülmektedir. Örneğin Deb vd. (2002), algoritma tarafından üretilen her bir nokta için baskın nokta kümesine olan en küçük Öklid uzaklığını ölçerek yakınsama hatasını bulmuş ve *ortalama yakınsama hatası*'nı performans ölçüsü olarak kullanmışlardır. $u(z, y) = \sqrt{\sum_{i=1}^{p}(z_i - y_i)^2}$ uzaklık ölçüsünün kullanıldığı ve $\frac{1}{|Z_T|}\sum_{y\in Z_T} \min_{z\in Z_B} u(z, y)$



Şekil 6. Epsilon Göstergesinin Hesaplanması

ile hesaplanan bu performans ölçüsü, temsili küme baskın noktalardan oluşuyorsa sıfır değerini almaktadır.

Van Veldhuizen (1999) ise temsili kümede yer alan ancak baskın nokta kümesinde yer almayan noktaları sayarak *hata oranı* olarak adlandırdığı bir ölçü tanımlamıştır. Benzer olarak, bulunabilen etkin çözümlerin/baskın noktaların oranı, *başarı oranı da*, $\frac{|Z_T \cap Z_B|}{|Z_B|}$ hesaplanabilir.

3.3 Dağılım Ölçüleri

Temsili küme bulunurken, kapsama özelliği kadar, kriter uzayına iyi dağılmış noktalar bulmak da önemlidir. Burada amaç, farklı bölgelerin eşit temsilini sağlamak ve birbirine çok yakın noktalar bulmayı engelleyerek az sayıda nokta ile iyi bir kapsama ölçüsüne ulaşmaktır. Sayın (2000) tarafından önerilen dağılım ölçüsü Tanım 8'de verilmiştir.

Tanım 8: Herhangi bir temsili alt küme $Z_T \in \mathbb{R}^p$ için, $u\ (y_1,\ y_2)\ y_1,\ y_2 \in Z_T$ arasındaki uzaklığı ve $\delta(Z_T) = \min_{\substack{y_1, y_2 \in Z_T \\ y_1 \neq y_2}} u(y_1, y_2)$ ise Z_T kümesinin *dağılım derecesini* göstermektedir.

Şekil 4, $u(y_1, y_2) = \max_{i=1,\dots,p} |y_{1i} - y_{2i}|$ uzaklık ölçüsü kullanıldığında verilen temsili küme için dağılım derecesini, $\delta(Z_T)$ göstermektedir.

Farklı olarak, Schott (1995), birbirine en yakın komşu temsili çözümler arasındaki uzaklıkların standart sapmasını dağılım ölçüsü olarak kullanmayı önerirken, Czyzżak ve Jaszkiweicz (1998) kapsama hatası ile ortalama temsil hatasını kullanarak farklı bir dağılım ölçüsü önermişlerdir. Ortalama temsil hatası $U1 = \frac{1}{|Z_B|} \sum_{z \in Z_B} \min_{y \in Z_T} u(z, y)$ ve kapsama hatası $U2 = \text{maks}_{z \in Z_B} \min_{y \in Z_T} u(z, y)$ olarak tanımlandığında, *dağılım oranı* $\frac{U2}{U1}$ kalite ölçüsü olarak kullanılmaktadır. Czyzżak ve Jaszkiweicz (1998), bu oran küçüldükçe eşit dağılımın arttığını belirtmişlerdir.

Zitzler (1999) ise temsili kümede yer alan herhangi bir noktadan verilen sabit bir uzaklık değerinden, σ, daha uzakta yer alan temsili nokta sayısının ortalamasını hesaplamıştır: $\frac{1}{|Z_T|-1} \sum_{y_1 \in Z_T} |\{y_2 \in Z_T : u(y_1, y_2) > \sigma\}|$

Srinivas ve Deb (1995), baskın nokta kümesini istenilen nokta sayısı kadar alt bölgeye ayırmışlardır. Daha sonra, algoritma tarafından üretilen noktaların her bir alt bölgeye düşen sayılarının ortalama ve standart sapması bulunarak *bölgesel dağılım derecesi*'ni hesaplamışlardır.

Deb vd. (2002), geliştirdikleri evrimsel algoritmanın performansını ortalama temsil hatasına ek olarak dağılım ölçüsü ile değerlendirmişlerdir. İki amaçlı problemler için kullanılan bu ölçüde, seçilen bir kritere göre ardışık olan temsili noktalar arasındaki uzaklık ölçülmekte, $u_t = u(y_t, y_{t+1})$ ve bu uzaklığın ortalaması, $\bar{u}$ hesaplanmaktadır. Daha sonra, her amaç fonksiyonu için baskın nokta kümesinde yer alan uç noktalar ile algoritma tarafından bulunan kümenin uç noktaları arasındaki Öklid uzaklıklar hesaplanarak toplamı, U_T, alınmaktadır. Temsili kümenin çeşitlilik derecesi:

$$\Delta(Z_T) = \frac{U_T + \sum_{t=1}^{|Z_T|} |u_t - \bar{u}|}{U_T + (|Z_T| - 1)\bar{u}}$$

olarak hesaplanmaktadır. Burada $\Delta(Z_T)$ küçüldükçe, eşit dağılımın arttığı belirtilmektedir.

Boland vd. (2016), temsili baskın noktaların dağılımını değerlendirmek için farklı bir ölçü kullanmışlardır. İlk olarak, her bulunamayan baskın nokta, $z \in Z_B \setminus Z_T$ için en yakınında olan temsilci belirlenmektedir. Daha sonra, her bir temsilcinin, ($y \in Z_T$), temsil ettiği baskın nokta sayısı $n(y)$ bulunarak, ortalama temsil edilen nokta sayısı $\tilde{\mu} = \frac{\sum_{y \in Z_T} n(y)}{|Z_T|}$ ve ortalamadan sapma $\tilde{\sigma} = \frac{\sum_{y \in Z_T} (n(y) - \tilde{\mu})^2}{|Z_T|}$ hesaplanmaktadır. Kalite ölçüsü olarak *dağılım göstergesi*, $\frac{\tilde{\sigma}}{\tilde{\mu}}$ kullanılmakta, bu oran küçüldükçe temsili kümenin daha iyi dağılıma sahip olduğu değerlendirilmektedir.

4. YAKLAŞIMLARIN İNCELENMESİ

Bu kısımda, temsili bir alt küme bulmayı amaçlayan ve ÇATP problemleri için uygulanabilir çalışmalar, metasezgisel/sezgisel yaklaşımlar ve yakınsama yaklaşımları olarak iki farklı grupta incelenecektir. İlk grupta, temsili kümenin büyüklüğünü kontrol edebilen ancak diğer kalite ölçüleri için teknikler kullanmakla beraber performans garantisi sunamayan metasezgisel/sezgisel yaklaşımlar yer almaktadır. Bu yaklaşımlar içerisinde, tüm baskın nokta kümesini bulmayı hedefleyen ancak çözüm zorluğu nedeni ile temsili küme bulan yaklaşımlar da incelenecektir. Daha sonra, istenilen kalite seviyesine sahip temsili küme bulan ve performans garantisi sunan yaklaşımlar yakınsama yaklaşımları başlığı altında tartışılacaktır. Bu kapsamda, ilgili çalışmalar çözüm metoduna göre sınıflandırılmakta; problem tipi, amaç fonksiyonu sayısı, ve kullanılan kalite ölçülerine göre incelenmektedir. Tablo 1, incelenen çalışmalar için özet bir liste sunmaktadır. Bu listede, ilk olarak özel yapıya sahip ÇATP problemleri olan ÇABO problemleri için kullanılan yöntemler sınıflandırılmaktadır. Daha sonra sırasıyla, ÇATP problemleri ve ÇATP problemlerini de kapsayan çok amaçlı karışık tamsayılı programlama (ÇAKTP) problemleri için geliştirilen çalışmalar listelenmektedir. Tablo 1'de en son olarak, daha genel ÇAO problemleri için geliştirilen ve ÇATP problemlerine de uygulanabilir yöntemler yer almaktadır.

4.1 Metasezgisel ve Sezgisel Yaklaşımlar

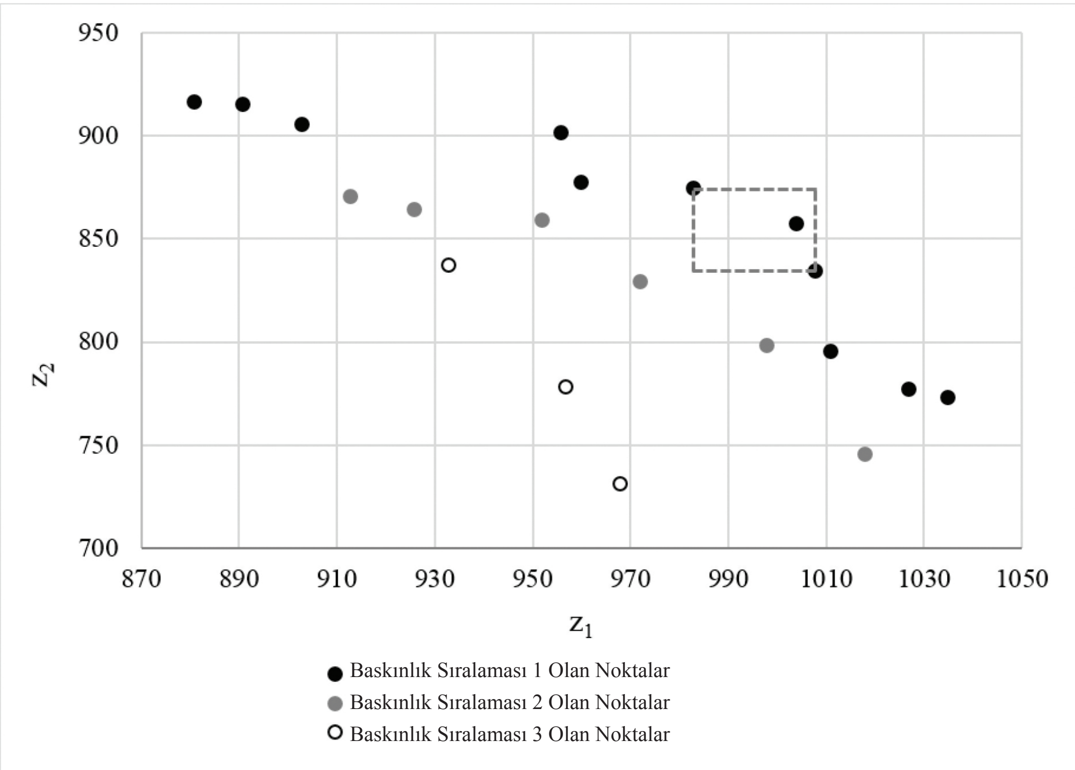
Problem büyüklüğü arttıkça baskın nokta sayısının hızla artması ve karşılaşılan çözüm zorlukları nedeni ile ÇATP problemleri için birçok metasezgisel ve sezgisel yöntem bulunmaktadır. Sezgisel yöntemler baskın noktaların yakınında iyi noktalar ararken, metasezgisel yöntemler ise birçok farklı probleme uyarlanabilen farklı metotları beraber kullanan bir çözüm konsepti sunmaktadır (Osman ve Laporte, 1996). Ehrgott ve Gandibleux (2008), ÇABO problemleri için geliştirilen hibrit metasezgisel yöntemler üzerine bir inceleme çalışması sunmuşlardır.

Çok amaçlı problemler için temsili küme bulmayı hedefleyen ilk çalışmalardan biri Armann (1987) tarafından önerilmiştir. Bu çalışmada, istenilen eleman sayısına ve iyi dağılıma sahip bir temsili küme bulmak amaçlanmaktadır. Her bir amaç fonksiyonunun ayrı ayrı optimize edilmesiyle elde edilen noktalar kullanılarak her amaç fonksiyonu için alt ve üst sınırlar tanımlanmaktadır. Toplam istenilen çözüm sayısı verildiğinde, her bir amaç fonksiyonu için, o amaç fonksiyonu en iyilenerek bulunacak çözüm sayısını hesaplamak için matematiksel bir model çözülmektedir. Belirlenen nokta sayısı kadar, o amaç fonksiyonu en iyilenerek belirlenen sınırlar içinde çözümler üretilmektedir. ÇATP problemleri için de uyarlamak mümkündür.

Metasezgisel yöntemlerin başında ise bir popülasyon üzerinde çalışarak birden fazla nokta üretebilen ve bu nedenle ÇAO için uyarlaması mümkün evrimsel algoritmalar (EA) bulunmaktadır. Bu kapsamda geliştirilen ve ÇATP problemleri için de uyarlanabilen birçok algoritma mevcuttur (Zhou vd., 2011). Mevcut EA'lar temel olarak tüm baskın nokta kümesini bulmayı hedeflemekte ve bu nedenle, temsili nokta kümesi bulmayı amaçlayan diğer algoritmalara kıyasla daha büyük bir popülasyon üzerinde çalışmaktadır. Burada, baskın noktalara yakın noktalar bularak baskın nokta kümesine yaklaşmak kadar, baskın nokta kümesininin farklı bölgelerini temsil etmek de önemlidir. Bunun için, hem kapsama hem dağılım ölçüleri performans ölçüsü olarak kullanılırken, popülasyon büyüklüğü ile istenilen çözüm sayısı kontrol edilebilmektedir. Bu nedenle, mevcut EA'ları popülasyon büyüklüğünü azaltarak baskın nokta kümesini iyi temsil eden nokta kümesi bulmak için de kullanmak mümkündür. Öyle ki, popülasyonda yer alan her birey, her nokta için *uygunluk fonksiyonu* değeri atanmakta ve bu değere göre bazı noktalar popülasyondan çıkarken bazıları hayatta kalmaya devam etmektedir. Shukla ve Deb (2007), iyi dağılmış temsili nokta kümesi bulmayı hedefleyen EA'lar ile klasik yöntemlerle matematiksel modeller çözerek nokta kümesi bulan yaklaşımları karşılaştırmışlardır.

Bu alandaki ilk çalışmalardan biri, Schaffer (1984) tarafından geliştirilen VEGA (Vector Evaluated Genetic Algorithm) yöntemidir. Bu yöntemde, nokta popülasyonu, her amaç fonksiyonu için ayrı bir alt popülasyon olacak şekilde alt popülasyonlara bölünmektedir. Her bir alt popülasyon için, ona karşılık gelen amaç fonksiyonunun optimize edilerek çözümler üretilmesi amaçlanmaktadır. Herhangi bir amaç fonksiyonunda çok iyi olan bireylerin korunduğu bu algoritma, herhangi bir amaç fonksiyonunda çok iyi olmayan ama farklı amaç fonksiyonlarında daha dengeli olan bireylerin hayatta kalmalarına imkan tanımaması nedeniyle tüm baskın nokta kümesinin temsili açısından iyi performans göstermemektedir.

Daha sonra geliştirilen yöntemlerde hem kapsama



Şekil 7. Baskınlık Sıralaması ve Kalabalıklık Uzaklığının Hesaplanması

hem de dağılım ölçüleri göz önünde bulundurulmuştur. Örneğin Goldberg (1989), *baskınlık sıralaması* yönteminin kullanılmasını önermektedir. Öyle ki, tüm baskın noktalar ilk sıraya yerleştirilirken, bu noktalar çıkarıldığında kalan popülasyona göre baskın olan noktalar ikinci sırada yer almaktadır. Şekil 7'de de gösterildiği gibi, diğer noktalar da baskınlık seviyesine göre sınıflandırılmakta ve uygunluk fonksiyonu değeri bu şekilde belirlenmektedir. Bu yöntem, Srinivas ve Deb (1995) tarafından geliştirilen NSGA (Nondominated Sorting Genetic Algorithm) tarafından da kullanılmaktadır.

Zitzler ve Thiele (1999) tarafından önerilen SPEA (Strength Pareto Evolutionary Algorithm) ise popülasyona göre baskın noktaları ve baskılanan noktaları ayrı birer popülasyon olarak saklamaktadır. Baskın olan noktalara, baskıladıkları nokta sayısına göre *dayanıklılık* değeri atayan SPEA, diğer noktalar için onları baskılayan noktaların dayanıklılık değerleri toplamını kullanmaktadır. Küçük dayanıklılık değerine sahip noktaların yer aldığı bölgelerde daha çok yeni nokta arayan SPEA, böylece iyi dağılıma sahip noktalar bulmayı amaçlamaktadır. Ayrıca, SPEA, üretilen nokta sayısını ve dağılımını kontrol edebilmek amacıyla, saklanan nokta sayısı belirli bir sınırın üzerine çıktığında kümeleme yöntemi ile üretilen noktaların bir kısmını elemektedir.

İyi bir dağılıma sahip nokta üretmeyi amaçlayan evrimsel yaklaşımlarda çokça kullanılan ve Goldberg ve Richardson (1987) tarafından da önerilen bir diğer yaklaşım ise *paylaşım fonksiyonu* hesaplamadır. Bu yaklaşımda, yeni bulunan bir noktanın mevcut noktalara olan uzaklığı hesaplanmakta ve belirli bir komşuluk tanımına göre çevresindeki çözüm yoğunluğu tahmin edilmektedir. En son olarak, uygunluk fonksiyonu değerleri çözüm yoğunluğuna göre atanmakta ve böylece çözüm yoğunluğu az olan bölgelerden yeni noktalar bulmak amaçlanmaktadır. Ancak, algoritmaların performansı, seçilen komşuluk tanımına bağlı olduğu için ve de her iki nokta için paylaşım fonksiyonunun hesaplanması gerektiği için zorluklara neden olmaktadır. Deb vd. (2002) ise NSGA ve SPEA'dan hem baskın noktalara yakın noktalar bulma hem iyi dağılıma sahip noktalar bulma açısından daha iyi performans gösteren NSGA-II yöntemini geliştirmişlerdir. Bu yöntemde paylaşım fonksiyonu yerine *kalabalıklık karşılaştırma* yaklaşımı kullanılmaktadır. Bu yaklaşımda ilk olarak, bir nokta etrafındaki nokta yoğunluğu hakkında bir tahmin yapabilmek için, Şekil 7'de de gösterildiği gibi, o noktanın her amaç fonksiyonunda her iki tarafında yer alan noktaların arasındaki ortalama uzaklık hesaplanmaktadır. Farklı amaç fonksiyonları için hesaplanan ve *kalabalıklık uzaklığı* olarak adlandırılan bu uzaklıklar toplanarak toplam kalabalıklık uzaklığı hesaplanmaktadır. Daha sonra, eşit bir dağılıma sahip bir nokta kümesi elde edebilmek için, aynı baskınlık seviyesine sahip noktalarda az kalabalık noktalar tercih edilmektedir.

Zitzler vd. (2002) ise SPEA algoritmasını geliştirerek SPEA2 algoritmasını önermişlerdir. SPEA2'de farklı olarak, dayanıklılık değeri hesaplanırken hem baskılayan noktaların sayısı hem de baskıladığı noktaların sayısı beraber dikkate alınmaktadır. Yoğunluk tahmini için, Silverman (1986) tarafından önerilen *h*. en yakın komşu yöntemi kullanılmaktadır. Öyle ki, *h*. en yakın komşuya olan uzaklığın tersi yoğunluk tahmini olarak değerlendirilmektedir. Daha sonra, hesaplanan ham uygunluk fonksiyonu değerine yoğunluk tahmini eklenerek uygunluk fonksiyonu değeri hesaplanlanmaktadır. Böylece, az yoğun bölgelere ayrıcalık tanınmaktadır. Çok amaçlı sırt çantası problemler üzerinde yapılan deneylerde diğer algoritmalardan iyi çalışan SPEA2 ve NSGA-II, benzer performans göstermektedir.

SPEA2 ve NSGA-II, bireylerin baskınlık seviyesine ve bulundukları bölgenin yoğunluk özelliklerine göre seçimler yapmaktadır. Farklı olarak, çok amaçlı optimizasyon problemleri için geliştirilen yöntemleri evrimsel algoritmalar ile beraber kullanan melez algoritmalar da bulunmaktadır. Ishibuchi ve Murata (1998), yerel arama yöntemlerini evrimsel algoritmalar ile beraber kullanan MOGLS (Multi-Objective Genetic Local Search Algorithm) yöntemini geliştirmişlerdir. MOGLS, uygunluk fonksiyonunu ağırlıklandırma yöntemi ile hesaplamakta ve yeni bulunan çözümün yakınında daha iyi çözümler için arama yapmaktadır. Her adımda, farklı ağırlıklar kullananan MOGLS, bu ağırlık vektörüne göre sadece belirli komşu bölgelerde arama yaparak çözüm zorluğunu azaltmaktadır.

Benzer şekilde, Zhang ve Li (2007), evrimsel algoritmaları ağırlıklandırma yöntemi ile birleştirerek melez bir yaklaşım olan MOEA/D (Multiobjective Evolutionary Algorithm based on Decomposition) yöntemini önermişlerdir. Bu yöntemde ilk olarak, problem alt problemlere bölünmekte ve daha sonra da bu alt problemlerin çözülmesi için evrimsel algoritmalar kullanılmaktadır. Her alt problemde, amaç fonksiyonlarının ağırlıklandırılmış toplamı uygunluk fonksiyonu olarak kullanılmakta ve yeni üretilen noktaların komşuluğunda daha iyi uygunluk fonksiyonu değerine sahip çözümler aranmaktadır. Farklı bölgelerden bireyler üretilmesi amacıyla her alt problem için farklı ağırlıklar kullanılmaktadır. Zhang ve Li (2007), çok amaçlı sırt çantası problemleri üzerinde yapılan deneylerde, kapsama göstergesi ve ortalama temsil hatası kalite ölçülerini kullanarak MOEA/D algoritmasının performansını MOGLS ve NSGA-II ile karşılaştırmışlardır.

Chen vd. (2017), başka bir melez yöntem olan DMOEA-εC (Decomposition-Based Multi-Objective Evolutionary Algorithm with the Epsilon-Constraint Framework) algoritmasını geliştirmişlerdir. Farklı olarak, bu algoritma alt problemi tanımlarken ağırlıklandırma yöntemi yerine epsilon kısıtı yöntemi kullanmakta ve her alt problem için farklı eşik değerleri belirlemektedir.

Zhang vd. (2016), evrimsel algoritmaları özdüzenleyici haritalar ile beraber kullanarak melez bir yaklaşım olan SMEA (Self-organizing Multi-objective Evolutionary Algorithm) yöntemini geliştirmişlerdir. Özdüzenleyici haritalar çok boyutlu uzayda yer alan noktaları, daha az boyutlu bir uzayda temsil eden noktalar üreten gözetimsiz öğrenme yöntemlerinden biridir (Kohonen, 1998). Her adımda, mevcut çözümler arasındaki komşuluk ilişkisi özdüzenleyici haritalar ile tanımlanmakta, her bir çözümün komşusu ile yeni çözüm üretmesine izin verilmektedir. Chen vd. (2017), DMOEA-εC, MOEA/D ve SMEA yöntemlerini çok amaçlı sırt çantası problemleri üzerinde karşılaştırarak ortalama temsil hatası, kapsama hatası ve üst hacim göstergesi değerlerini raporlamışlardır. Yapılan deneyler, DMOEA-εC yöntemininin diğerlerinden daha iyi kalitede çözümler bulduğunu göstermektedir.

Evrimsel algoritmalara ek olarak, çok amaçlı problemler için geliştirilmiş diğer metasezgisel ve sezgisel yöntemler de bulunmaktadır.

Czyżżak ve Jaszkiweicz (1998), ÇABO problemleri için yaklaşık olarak bir baskın küme bulan metasezgisel bir yöntem önermişlerdir. PSA (Pareto Simulated Annealing, Pareto Benzetilmiş Tavlama) olarak adlandırılan bu algoritmada, her çözümün komşuluğunda yeni çözümler aranmakta ve amaç fonksiyonlarının ağırlıkları ayarlanarak baskın nokta kümesine yaklaşılırken eşit aralıklarla iyi dağılmış bir küme bulmak amaçlanmaktadır. Kalite ölçüsü olarak, farklı eleman sayısı ölçüsüne sahip temsili kümeler için ortalama temsil hatası ve kapsama hatası kullanılmıştır. Benzer yöntem ve kalite ölçülerini kullanan Ulungu vd. (1999), ÇABO problemleri için baskın nokta kümesini yaklaşık olarak bulan MOSA (Multiobjective Simulated Annealing) yöntemini önermişlerdir. Çok amaçlı sırt çantası problemi için uygulanan bu yöntemi ön bir çalışmadan sonra diğer ÇABO problemlerine de uyarlamanın mümkün olduğu belirtilmektedir. Tuyttens vd. (2000), MOSA'yı açgözlü bir yaklaşım ile geliştirerek çok amaçlı atama problemine uyarlamışlardır.

Karasakal ve Köksalan (2009) ise ilk olarak, baskın nokta kümesini yaklaşık olarak tahmin ederek istenilen kalitede bir nokta kümesi belirledikten sonra gerçek baskın noktalardan oluşan temsili küme bulmuşlardır. Bu algoritmada, birbirinden hemen hemen eşit uzaklıkta, iyi dağılıma sahip temsili noktalar bulmak amaçlanmakta ve bu nedenle ilk olarak, baskın nokta kümesini yaklaşık olarak temsil eden bir yüzey tanımlanmaktadır. Yüzeyin tanımlanmasında ilk olarak, Köksalan (1999) tarafından önerilen ve daha sonra Köksalan ve Lokman (2009) tarafından ÇABO problemleri üzerinde iyi çalıştığı gösterilen L_q yüzeyi kullanılmaktadır:

$$L_q(z_1, \ldots, z_p) = \left(\sum_{k=1}^{p} \lambda_k (z_k - z_k^N)^q \right)^{1/q} = d$$

Burada λ, negatif olmayan ağırlık vektörünü göstermekte ve d değerinin ölçeklendirilmesinde de kullanılabileceği için değeri 1 olarak kullanılmaktadır. Her problem için belirli sayıda baskın nokta üreterek o noktalardan geçen en yakın ağırlıklandırılmış L_q yüzeyini tanımlamaktadır. Daha sonra, tanımlanan bu yüzey üzerinde eşit dağılmış hipotetik noktalar üretilmekte ve bu noktalar referans alınarak *başarı skalarlaştırma* *programları* (ASP-Achievement Scalarizing Program) çözülerek gerçek baskın noktalar bulunmaktadır (Wierzbicki, 1980):

(ASP)

$$\text{Min } \alpha - \sum_{k=1}^{p} \rho_k z_k(\text{x})$$

Kısıtlar

$$\alpha \geq \left(r_k - z_k(\text{x})\right)\frac{1}{g_k} \qquad k = 1, \dots, p$$

$$\text{x} \in \text{X}$$

Burada *r*, referans noktasını; *g* ise *r* noktasındaki gradyant vektörünü göstermektedir. Algoritmanın performansı eleman sayısı, kapsama hatası ve dağılım ölçülerine göre değerlendirilmiştir. Çok amaçlı doğrusal programlama problemleri üzerinde test edilen bu algoritmayı ÇATP problemleri için de uyarlamak mümkündür.

Aytuğ ve Sayın (2009), daha önceki çalışmalardan farklı olarak DVM (Destekçi Vektör Makinesi/Support Vector Machine) kullanarak baskın noktaların yerlerini öğrenmeye çalışan bir yöntem önermişlerdir. Sınıflandırma problemlerinde, farklı sınıfları birbirinden ayıran bir düzlem bulmaya çalışan DVM'de her iki sınıfa en uzakta olan düzlemi bulmak amaçlanır. Doğrusal bir düzlem ile ayrılamayan sınıflar için, doğrusal olmayan bir dönüşüm uygulanarak farklı uzayda bir ayırım yapmak da mümkündür. Bu çalışmada DVM, başlangıç nokta kümesi verilerek ÇATP problemleri için baskın ve baskılanan noktaları ayırmak için kullanılmaktadır. Daha sonra bu sınıflandırma, çok amaçlı evrimsel bir algoritma içinde kullanılarak temsili bir nokta kümesi üretmekte kullanılmaktadır. Çok amaçlı sırt çantası ve çok amaçlı atama problemleri üzerinde yapılan deneylerde ortalama temsil hatası, kapsama hatası ve yakınsama hatası ölçülmektedir.

Mavrotas ve Florios (2013), epsilon-kısıtı yöntemini geliştirerek ÇATP problemleri için AUGMECON2 algoritmasını tasarlamışlardır. Epsilon-kısıtı yönteminde kullanılan ($P^{\varepsilon j}$) modelinden farklı olarak, *k*. amaç fonksiyonu değerinin ε_k değerinin üstünde kalan miktarı, s_k $k \neq j$ amaç fonksiyonuna dahil edilmiştir:

$(P^{AUGMECON2})$

$$\text{Maks } z_j(\text{x}) + \sum_{\substack{k=1 \\ k \neq j}}^{p} \rho_k s_k$$

Kısıtlar

$$z_k(\text{x}) - s_k = \varepsilon_k \quad k = 1, \dots, p \quad k \neq j$$

$$\text{x} \in \text{X}$$

Bu algoritmada, her amaç fonksiyonunun alabileceği değer aralığı q_k tane eşit aralığa bölünerek amaç fonksiyonu uzayı küçük parçalara ayrılmaktadır. Her iterasyonda, daha önce bulunan çözümlerdeki s_k değişkenlerin değerleri ve çözüm uzayında bulundukları bölgeler dikkate alınarak ε_k değerleri belirlenmektedir. Tüm baskın noktaları bulmak için tasarlanan bu algoritma aynı zamanda tüm baskın noktaları yaklaşık olarak temsil eden bir küme bulmak için de modifiye edilmiştir. Çözüm uzayı farklı sayıda parçalara bölünerek nokta sayısı farklı temsili çözüm kümeleri üretilmiş, bu kümelerin kalitesi tüm baskın nokta kümesi ile karşılaştırılarak kapsama göstergesi, $C\ (Z_{\text{AUGMECON2}}, Z_B)$ raporlanmıştır. Ayrıca, SPEA2'nin iyi çalıştığı gösterilen sırt çantası problemleri üzerinde deneyler yapılmış ve iki nokta kümesinini birbirine göre dağılımını ölçen kapsama göstergesi, $C\ (Z_{\text{SPEA2}}, Z_{\text{AUGMECON2}})$ raporlanmıştır.

Faulkenberg ve Wiecek (2012), iki amaçlı programlama problemleri için temsili küme bulan iki yöntem önermişlerdir. İlk yaklaşımda, epsilon-kısıtı yönteminin bir varyantı olan *kısıt-kontrollü aralıklara bölme* yöntemi kullanılmıştır. Öyle ki, bu yöntemde bilinen bir $z\ (x^*)$ noktası ile istenilen aralığa sahip bir nokta bulmak için (P^{ϕ}) alttaki modeli çözümlenmiştir.

(P^{ϕ})

$$Maks\ z_1\ (x)$$

Kısıtlar

$$z_2\ (x) \geq z_2\ (x^*)$$

$$u\ (z(x), z(x^*)) \geq \phi$$

$$\text{x} \in \text{X}$$

Burada ϕ ise istenilen aralık değerini göstermektedir. Bu yöntem genelleştirilerek, istenilen aralıklara sahip temsili küme bulan bir yöntem geliştirilmiştir. ϕ değeri istenilen çözüme göre seçilmektedir. Diğer önerilen yöntemde ise iki seviyeli formülasyona dayalı bir aralık kontrol yöntemi önerilmektedir. İlk yönteme benzer olarak, daha önce bulunan noktalar referans alınarak verilen iki noktadan eşit uzaklıkta yeni bir nokta bulunması amaçlanmaktadır. Alt seviyedeki problem, baskın nokta bulmak ve üst seviyedeki problem ise aralığı kontrol etmek için çözülmektedir. İki amaçlı programlama problemleri üzerinde deneyler yapılan bu iki yöntemi, ÇATP için uyarlamak mümkündür.

Wang (2015), ÇATP ve ÇAKTP problemleri için doğrudan zigzag arama (DZZ) yöntemini önermiştir. Amaç fonksiyonunun *kara-kutu* olarak tanımlandığı ve bu nedenle karar değişkenlerine bağlı çalıştırılan benzetim modeli ile değerlendirildiği gerçek hayat problemlerine uygun bu yöntemde, zigzag ilerleyen baskın noktalar aranmaktadır. Tüm baskın nokta kümesini bulmayı amaçlayan bu yaklaşımda temsili bir küme bulunmakta ve performansı NSGA-II yöntemi ile kıyaslanmaktadır. Performans ölçüsü olarak ortalama yakınsama hatası ve bulunabilen baskın noktaların oranı raporlanmıştır.

4.2 Yakınsama Yaklaşımları

Metasezgisel ve sezgisel yöntemler, belirli kalite ölçülerine göre baskın kümeyi iyi temsil eden baskın veya baskılanan noktalardan oluşan bir küme bulmayı amaçlamakta ve bunun için farklı teknikleri bir arada kullanmaktadır. İstenilen çözüm sayısı bu algoritmalarda kontrol edilebilmekte, ancak diğer kalite ölçüleri çoğu zaman algoritmaların performansını değerlendirmekte kullanılmaktadır. Farklı olarak yakınsama yaklaşımları, istenilen kalite seviyesine sahip temsili küme üretmeyi amaçlamakta ve performans garantisi sunmaktadır.

Bu kapsamda, istenilen epsilon göstergesi değerine sahip en küçük temsili nokta kümesi veya istenilen eleman sayısı ile en küçük epsilon göstergesi değerine sahip temsili küme yaratmayı amaçlayan birçok yaklaşım geliştirilmiştir.

Ruhe ve Fruhwirth (1990), Burkard vd. (1991), tek amaçlı optimizasyon problemleri için alt ve üst sınırları sıkıştırarak istenilen epsilon göstergesine sahip çözüm bulan *sandviç* algoritmasını iki kriter için uyarlamışlardır. Minimum maliyetli akış problemleri üzerinde deneyler yapılmıştır.

Vassilvitskii ve Yannakakis (2005) ise üretilen çözüm sayısı üzerine performans garantisi veren açgözlü bir yaklaşım önermişlerdir. Verilen bir epsilon göstergesi (ε) değerini sağlayacak en küçük temsili kümenin eleman sayısı, opt_ε olarak tanımlanırsa, en fazla $3opt_\varepsilon$ sayıda çözüm ile istenilen epsilon göstergesi değerine ulaşılabilmektedir. Bu algoritmaya iyileştirmeler önererek farklı problemlere uygulanabilir. $2opt_\varepsilon$ performans garantisine sahip bir algoritma tasarlayan Diakonikolas ve Yannakakis (2009), iki amaçlı en kısa yol ve kapsayan ağaç problemi için Vassilvitskii ve Yannakakis'e (2005) göre daha iyi bir performans garantisi sunmuşlardır. Bazgan vd. (2015), bu çalışmaları inceleyerek tüm iki amaçlı optimizasyon problemleri için uygulanabilir yeni bir açgözlü yaklaşım, $3opt_\varepsilon$ tasarlamışlardır. Daha fazla amaç fonksiyonu olduğunda performans garantisi veren bir algoritma tasarlamanın zor olduğunu belirten Bazgan vd. (2015), ancak tüm noktalar verildiğinde istenilen epsilon göstergesine sahip en küçük alt küme bulmak için bir çalışma önermişlerdir.

Ponte vd. (2012), dal ve sınır yönteminin bir uyarlaması olan ışın araması yaklaşımını ÇABO problemleri için geliştirmişlerdir. İki amaçlı sırt çantası problemi üzerinde deneyler yapan Ponte vd. (2012), verilen bir epsilon göstergesi değerine sahip en küçük alt küme bulmayı amaçlamışlardır.

Filippi ve Stevanato (2013), iki amaçlı ÇABO problemleri için yaklaşık olarak temsili alt küme bulan iki yöntem önermişlerdir. İlk algoritmada, her adımda elde edilen bir baskın nokta ile çözüm uzayındaki bölge dört parçaya ayrılmaktadır. Baskılanan ve baskılayan parçaların dışında kalan iki parçada arama devam etmektedir. Bu arama yönteminde, izin verilen hata payına göre bir önceki çözüme yakın bazı bölgeler de arama uzayından çıkarılmaktadır. İkinci yöntemde ise epsilon-kısıtı yönteminin bir modifikasyonu uygulanarak izin verilen hata payına göre baskın nokta kümesini yaklaşık olarak temsil eden bir nokta kümesi üretilmektedir. Kar ve maliyet amaç fonksiyonlarına sahip seyyar satıcı problemi üzerinde test edilen bu yöntemler, epsilon göstergesi açısından performans garantisi sunmaktadır. Ayrıca, bulunabilen baskın noktaların sayısı da raporlanmıştır.

Pospelov (2009), amaç fonksiyonu uzayında çok yüzlü baskın konveks kabuğunu yaklaşık olarak bulan bir metot önermiştir. ÇATP problemleri için uygulanabilir olan bu yöntemde, dal ve sınır yöntemi ile konveks kabuk bulmak için kullanılan yöntemler birlikte kullanılmıştır. Kabul edilebilir hata payına göre, konveks kabuğun her bir yüzeyi yaklaşık olarak bulunmaktadır. Bu yöntemde, konveks kabuk üzerinde yer alan destekli baskın noktalar bulunurken, desteksiz baskın noktaların da olası yerlerini KV'ye sunmak mümkündür.

Sayın ve Kouvelis (2005), iki amaçlı tamsayılı programlama problemleri üzerinde çalışmışlar ve iki aşamalı bir yaklaşım önermişlerdir. İki amaçlı problemler için tüm baskın noktaları bulabilen bu algoritma, istenen kalite ölçülerine sahip bir alt küme bulunması için de adapte edilmiştir. Bu algoritmanın ilk aşamasında, verilen bir ağırlık vektörü, $\mathrm{w} \geq 0$ için $\mathrm{z}^{P_w} = \mathrm{maks}_{\mathrm{x}\in\mathrm{X}} \min_{k=1,..,p} w_k z_k(\mathrm{x})$ altproblemi çözülmekte ve ikinci aşamasında ise $\underset{\substack{\mathrm{x}\in\mathrm{X} \\ w_k z_k(\mathrm{x}) \geq \mathrm{z}^{P_w}}}{\mathrm{maks}} \sum_{k=1}^{p} w_k z_k(\mathrm{x})$ çözülerek olası baskın olmayan noktalar elenmektedir. Farklı ağırlık vektörleri kullanılarak iki amaçlı tamsayı problemleri için tüm baskın noktalar bulunabildiği gibi, her bir kriter aralığına göre yüzde cinsinden tolere edilebilecek kapsama hatası oranı verildiğinde, istenen kalitede temsili alt küme bulmak da mümkündür. Kouvelis ve Sayın (2006), bu algoritmayı hem çözümsel hem de kalite ölçüleri açısından geliştirerek sezgisel bir varyantını tasarlamış, sırt çantası, atama, minimum maliyetli ağ problemleri gibi çok çalışılan iki amaçlı ÇATP problemlerine uygulamışlardır.

Hamacher vd. (2007), iki amaçlı ÇATP problemleri için istenilen kalite ölçülerine göre temsili küme bulabilen *kutu yöntemini* geliştirmişlerdir. Her bir noktanın, bir kutu ile ilişkilendirilerek o kutudaki her noktayı temsil ettiği kabul edilmiştir. İstenilen kapsama hatasına ve istenilen nokta sayısına göre kutuların büyüklüğü tanımlanarak epsilon-kısıtı metodunun bir varyantı ile temsili alt küme üretilmektedir.

Sylva ve Crema (2007) ve Masin ve Bukchin (2008), ÇATP ve ÇAKTP problemleri için temsili baskın nokta kümesi bulan algoritmalar geliştirmişlerdir. Bu algoritmalarda, her adımda mevcut baskın nokta kümesi tarafından en kötü temsil edilen, diğer bir deyişle, en fazla temsil hatası yapılan baskın nokta bulunmaktadır. Bunun için, mevcut n baskın nokta, $z_t=(z_{t1},\ldots,z_{tp})$ $t=1,\ldots,n$ ise bu kümeden en uzakta yer alan yeni bir baskın nokta, z_{n+1}, bulan bir model çözülmektedir. Her iki çalışmada benzer matematiksel modeller kullanılmıştır. Model (P_n)'de gösterildiği gibi, baskın nokta kümesinden olan uzaklığı temsil eden α değeri maksimize edilerek en uzakta yer alan noktayı bulmak amaçlanmaktadır. Optimal α değeri için $\alpha^* = \mathrm{maks}_{\mathrm{x}\in\mathrm{X}_E} \min_{t=1,\ldots,n} \mathrm{maks}_{k=1,..,p} (z_k(\mathrm{x}) - z_{tk})$ yazılabilir.

(P_n)

$$\text{Maks } \alpha + \sum_{k=1}^{p} \rho_k\, z_k(\mathrm{x})$$

Kısıtlar

$$z_k(\mathrm{x}) \geq z_{tk} y_{tk} + \alpha - (M_k + U)(1 - y_{tk}) \quad k = 1, \ldots, p \quad t = 1, \ldots n \tag{1}$$

$$\sum_{k=1}^{p} y_{tk} \geq 1 \qquad t = 1, \ldots, n \tag{2}$$

$$\alpha \geq 0$$

$$y_{tk} \in \{0,1\}$$

$$\mathrm{x} \in \mathrm{X}$$

Model (P_n)'de $\rho_k>0$, baskın nokta bulmayı garantilemek amacıyla yeterince küçük seçilen pozitif bir sabit değeri temsil etmektedir. Kısıt (1)'de yer alan, $-\mathrm{M}_k$, $z_k(x)$ için alt sınır değerine (öyle ki $z_k(x) \geq -M_k \;\forall\; x\in X$), U ise herhangi iki baskın nokta arasındaki Tchebycheff uzaklık için üst sınır değere karşılık gelmektedir: $U \geq \max_{z_1,\, z_2 \,\in\, Z_B} \max_{k=1,..,p} |z_{1k} - z_{2k}|$

Her adımda en fazla kapsama hatasına sahip, diğer bir deyişle, en kötü temsil edilen baskın nokta bulunması nedeniyle, kapsama hatasının algoritma boyunca artmadan ilerleyeceği garantilenmektedir. Burada, Masin ve Bukchin'in (2008) çalışmasından farklı olarak, amaç fonksiyonu değerleri için ölçeklendirme yapılmış, kullanılan uzaklık ve kapsama ölçüsü sırasıyla, $u(z, y) = \max_{k=1,..,p} \frac{z_k - y_k}{z_k^I - z_k^N}$ ve

$\alpha(Z_T) = \max_{z \in Z_B} \min_{y \in Z_T} u(z, y)$ olarak tanımlanmıştır. Geliştirilen algoritmalarda KV, istenilen baskın nokta sayısına veya istenilen kapsama hatası seviyesine gelince algoritmayı durdurabilir. Bu nedenle, bu algoritma ile ÇATP problemleri için tüm baskın nokta kümesini, ÇAKTP problemleri için ise baskın noktaların bir kısmını bulmak da mümkündür. Ancak, bu algoritmalarda her yeni baskın nokta için model (P_n)'de de gösterildiği gibi, p tane ikili değişken ve p+1 tane kısıt eklenmesi gerekmektedir. Bu da problem büyüklüğü ve amaç sayısına bağlı olarak temsili küme bulunmasında çözüm zorluğuna neden olmaktadır.

Ceyhan (2014) ÇATP problemleri için temsili baskın nokta kümesi üreten çeşitli algoritmalar geliştirmiştir. İlk olarak, Sylva ve Crema (2007) ve Masin ve Bukchin (2008) tarafından geliştirilen algoritmaları revize ederek istenilen kapsama hatası ya da istenilen çözüm sayısına sahip temsili baskın nokta kümesi üreten bir algoritma tasarlamıştır. Bu algoritmada, Lokman ve Köksalan (2013) tarafından geliştirilen bir ayrıştırma ve arama algoritması ile model (P_n)'in olurlu kümesi daha küçük alt bölgelere bölünerek, ek ikili değişkenler ve kısıtlar kullanılmadan tanımlanmaktadır. Bu sayede, model (P_n) yerine birden fazla küçük modeller çözülmekte ve çözüm süresi önemli ölçüde azalmaktadır. İkinci algoritmada ise tolere edilebilecek kapsama hatası, α*, algoritma başında verildiği varsayımı altında, bulunan baskın noktaların tolere edilebilecek uzaklığında olan noktaların olurlu kümeden çıkarılmasını sağlayan α≥α* kısıtı modele dahil edilmiştir. Bu algoritma ve daha önce tasarlanan algoritmalar, KV tarafından istenilen çözüm bilgisi algoritmaya dahil edilmeden ilerleyebilen ve istenilen kalite seviyesine ulaştığında durdurulabilen algoritmalardır. İstenilen çözüm sayısının bilindiği varsayımı altında tasarlanan üçüncü yaklaşımda, öncelikle baskın noktaların olası yerleri Köksalan ve Lokman (2009) tarafından önerilen L_q yüzeyi ile yaklaşık olarak tahmin edilmektedir. Bu yüzey üzerinde istenilen çözüm sayısı kadar hipotetik nokta kümesi bulunmakta ve en son olarak bu noktalara en yakın gerçek baskın noktalar başarı skalarlaştırma programları çözülerek üretilmektedir.

Eusebi vd. (2014), iki amaçlı ağ akışı problemleri için temsili baskın nokta kümesi bulan bir algoritma tasarlamışlardır. Bir dizi epsilon-kısıtı problemi, dal ve sınır yöntemi ile çözülerek kapsama hatası veya dağılım ölçüsüne sahip temsili baskın nokta alt kümesi bulunmaktadır. Her adımda, yeni bir baskın nokta kümesi bulan bu yöntemde istenilen kalite seviyesine geldiğinde durdurulmaktadır.

Vaz vd. (2015), iki amaçlı ÇATP problemleri için temsili küme bulan algoritmalar tasarlamışlardır. Bu algoritmalarda, istenilen eleman sayısı verildiğinde, sırasıyla maksimum dağılım derecesine, minimum kapsama hatasına ve minimum epsilon göstergesine sahip temsili alt kümeler bulmak amaçlanmaktadır. İstenen kalite ölçülerine göre özel yapılandırılmış tesis yerleştirme problemleri olarak modellenen bu algoritmalarda, dinamik programlama ve eşik değer (threshold) yaklaşımları önerilmiştir. Dinamik programlamada, bir dizi küçük optimizasyon problemi çözülürken, eşik değer yaklaşımında optimal çözüm bulunana kadar bir dizi olurluluk problemi çözülmektedir. Bu çalışmada, belirtilen üç kalite ölçüsüne göre polinom zamanda çözülebilen iki amaçlı ÇABO problemleri dikkate alınmıştır. Bu çalışmayı, daha çok amaç fonksiyonuna sahip problemler için genelleştirmek kolay değildir.

Boland vd. (2015), iki amaçlı ÇATP için arama uzayını kutulara bölerek ilerleyen dengeli bir kutu yöntemi (Balanced Box Method/BBM) önermişlerdir. Bu yöntemde, yeni bir çözüm bulunmadan önce arama uzayı iki eşit parçaya ayrılmaktadır. Daha sonra, bir amaç fonksiyonu en iyilenerek yeni bir nokta bulunmakta ve bulunan noktalar kullanılarak kutular daha da daraltılmaktadır. İki amaçlı tamsayı programları için tüm baskın noktaları bulan BBM, tüm baskın nokta kümesini temsil eden bir alt küme bulmak için de tasarlanmıştır. Burada, belirli bir üst hacim boşluğu değerine ulaşabilmek için ürettiği baskın nokta sayısı performans ölçüsü olarak kullanılmaktadır.

Boland vd. (2016), üç amaçlı tamsayı programları için tüm baskın nokta kümesini bulan bir arama yöntemi, LSM (L-Shape Search Method) geliştirmişlerdir. Her adımda, yeni bulunan noktaların baskıladığı bölgeler kullanılarak kriter uzayı L-şeklinde arama bölgelerine ayrılmaktadır. Her adımda arama bölgeleri daraltılarak tüm baskın noktalar bulunmaktadır. Ek olarak, Boland

Tablo 1. Sınıflandırma

Problem Tipi	Amaç Sayısı	Yaklaşım	Çözüm Metodu	Kullanılan Kalite Ölçüleri	Çalışmalar
ÇABO	*2*	Yakınsama	Dal ve sınır yöntemi Işın araması	Eleman sayısı Epsilon göstergesi	Ponte vd. (2012)
		Yakınsama	Epsilon-kısıtı metodu Kutu yöntemi	Eleman sayısı Epsilon göstergesi	Filippi ve Stevanato (2013)
		Yakınsama	Epsilon-kısıtı metodu Dal ve sınır yöntemi	Eleman sayısı Kapsama hatası Dağılım derecesi	Eusebi vd. (2014)
		Yakınsama	Dinamik programlama Eşik değer algoritması	Eleman sayısı Kapsama hatası Epsilon göstergesi Dağılım derecesi	Vaz vd. (2015)
	p	Sezgisel/ Metasezgisel	Benzetilmiş tavlama	Kapsama hatası Ortalama temsil hatası Başarı oranı Dağılım oranı	Czyżżak ve Jaszkiweicz (1998) Ulungu vd. (1999) Tuyttens vd. (2000)
ÇATP	*2*	Yakınsama	İki aşamalı optimizasyon yaklaşımı	Eleman sayısı Kapsama hatası	Sayın ve Kouvelis (2005) Kouvelis ve Sayın (2006)
		Yakınsama	Epsilon-kısıtı metodu Kutu yöntemi	Eleman sayısı Kapsama hatası	Hamacher vd. (2007)
		Yakınsama	Uzayı bölme ve arama yöntemi	Eleman sayısı Üsthacim boşluğu	Boland vd. (2015)
	3	Yakınsama	Uzayı bölme ve arama yöntemi	Başarı oranı Üsthacim boşluğu Kapsama hatası Ortalama temsil hatası Dağılım göstergesi	Boland vd. (2016)
	p	Sezgisel/ Metasezgisel	Destekçi vektör makinesi Evrimsel algoritma	Kapsama hatası Ortalama temsil hatası Ortalama yakınsama hatası	Aytuğ ve Sayın (2009)
		Sezgisel/ Metasezgisel	Epsilon-kısıtı metodu	Eleman Sayısı Kapsama göstergesi	Mavrotas ve Florios (2013)
		Yakınsama	Yüzey tanımlama Konveks Kabuk	Eleman sayısı Ortalama yakınsama hatası	Pospelov (2009)
ÇAKTP	*p*	Sezgisel/ Metasezgisel	Arama yöntemi	Başarı oranı Ortalama yakınsama hatası	Wang (2015)
		Yakınsama	Uzayı bölme ve arama yöntemi	Eleman sayısı Kapsama hatası	Sylva ve Crema (2007) Masin ve Bukchin (2008) Ceyhan (2014)
ÇAO	*2*	Sezgisel/ Metasezgisel	Epsilon-kısıtı metodu	Eleman sayısı Dağılım derecesi	Faulkenberg ve Wiecek (2012)
		Yakınsama	Sandviç algoritması	Eleman sayısı Epsilon göstergesi	Ruhe ve Fruhwirth (1990)
	p	Sezgisel/ Metasezgisel	Uzayı bölme ve arama yöntemi	Eleman sayısı	Armann (1987)
		Sezgisel/ Metasezgisel	Yüzey tanımlama Başarı skalarlaştırma programı	Eleman sayısı Kapsama hatası Dağılım derecesi	Karasakal ve Köksalan (2009)
		Sezgisel/ Metasezgisel	Evrimsel algoritma	Eleman sayısı	Schaffer (1984)
				Eleman sayısı Bölgesel dağılım derecesi	Srinivas ve Deb (1995)
				Eleman sayısı Kapsama göstergesi Üst hacim göstergesi	Zitzler ve Thiele (1999)
				Eleman sayısı Üst hacim göstergesi	Zitzler vd. (2002)
				Eleman sayısı Ortalama yakınsama hatası Çeşitlilik derecesi	Deb vd. (2002)
		Sezgisel/ Metasezgisel	Evrimsel algoritma Yerel arama yöntemi	Başarı oranı	Ishibuchi ve Murata (1998)
		Sezgisel/ Metasezgisel	Evrimsel algoritma Ağırlıklandırma yöntemi	Kapsama göstergesi Ortalama temsil hatası	Zhang ve Li (2007)
		Sezgisel/ Metasezgisel	Evrimsel algoritma Özdüzenleyici haritalar	Ortalama temsil hatası Üst hacim göstergesi	Zhang vd. (2016)
		Sezgisel/ Metasezgisel	Evrimsel algoritma Epsilon-kısıtı metodu	Ortalama temsil hatası Kapsama hatası Üst hacim göstergesi	Chen vd. (2017)
		Yakınsama	Açgözlü yaklaşımı	Eleman sayısı Epsilon göstergesi	Vassilvitskii ve Yannakakis (2005) Diakonikolas ve Yannakakis (2009) Bazgan vd. (2015)

vd. (2016), tüm baskın noktaları bulmak için gerekli sürenin onda biri süre sınırı koyarak mevcut algoritma ile temsili bir baskın nokta kümesi bulmayı önermişlerdir. Performans ölçüsü olarak, nadir noktası referans alınarak üst hacim boşluğunun yüzdesi bulunmaktadır. Ayrıca, bulunan noktaların yüzdesi, kapsama hatası ve ortalama temsil hatası kalite ölçüsü olarak kullanılmaktadır. Dağılımı değerlendirmek için ise dağılım göstergesi hesaplanmaktadır.

5. SONUÇ

Farklı sektörlerde yaygın olarak karşılaşılan ÇATP problemleri için genellikle birçok anlamlı çözüm vardır. Değişik çözümler, değişik amaçlar öne çıktığında anlam kazanabilir ve değişik karar vericiler için farklı çözümlerin tercih edilmesi doğaldır. Genellikle baskın nokta sayısı çok fazladır ve hepsini bulmak hem çözüm zorluğu hem de değerlendirme yapmak açısından uygulanabilir bir yaklaşım değildir. Bu nedenle, baskın noktaları iyi temsil eden, istenilen kalitede küçük alt kümeler bulmak önemlidir.

Bu çalışmada ilk olarak, temsili kümenin performansını değerlendirmek amacıyla önerilen kalite ölçüleri eleman sayısı, kapsama ve dağılım ölçüleri başlıkları altında sınıflandırılarak incelenmiştir. Daha sonra, istenilen kalite ölçüleri dikkate alarak temsili alt küme bulan ve ÇATP problemleri için uygulanabilir yaklaşımlar tartışılmıştır. İki amaçlı optimizasyon problemleri, basitleştirici özellikleri nedeniyle çok amaçlı problemlerden ayrı olarak listelenmiştir. Bu yaklaşımları çok amaçlı problemler için genellemek çok kolay değildir.

İncelenen çalışmalarda, baskın noktalara yakın noktalar bulmak kadar, iyi dağılıma sahip, kriter uzayının farklı bölgelerini eşit oranda temsil eden nokta kümesi bulmanın önemi vurgulanmıştır. ÇATP problemlerinin çözüm zorluğu düşünüldüğünde, istenilen kalite seviyesinde minimum eleman sayısına sahip veya istenilen eleman sayısı ile en iyi kalite seviyesine sahip alt küme bulmak daha da önem kazanmaktadır. Bu kapsamda geliştirilen yöntemlerin çoğunda art arda matematiksel modeller çözülerek istenilen özelliklere sahip alt kümeyi oluşturacak yeni noktalar adım adım aranmaktadır.

Ancak, problem büyüklüğü arttıkça çözüm zorluğu artmakta, bu da metasezgisel yöntemlerin ön plana çıkmasına neden olmaktadır. Özellikle, popülasyon bazlı çalışması nedeni ile birden çok çözüm üretebilen evrimsel algoritmalar çok amaçlı optimizasyon problemlerine çok uygundur. İstenilen kalite ölçüleri, populasyondaki bireylerin, yani noktaların uygunluk fonksiyonu değerlerinin hesaplanmasında dikkate alınmaktadır. Böylece, popülasyonda istenilen kalite ölçülerine göre bazı noktalar elenirken, bazı noktalar popülasyonda kalma şansına sahip olmaktadır. Bu algoritmalarda istenilen kalite seviyesine ulaşmak için farklı teknikler kullanılsa da kapsama ve dağılım ölçüleri için performans garantisi sunulamamaktadır.

Mevcut çalışmalar incelendiğinde, ÇATP problemleri için iyi temsili küme bulmak, geliştirilen birçok çalışma olmasına rağmen, çözüm zorlukları veya sundukları kalite seviyeleri nedeniyle üzerine çalışılması halen önem taşıyan alanlardan biridir. Burada, sadece baskın noktalardan oluşan temsili küme bulan yaklaşımlar yerine, baskın kümesine kabul edilebilecek uzaklıkta yer alan noktalardan oluşan bir küme bulan pratik yaklaşımlar önemlidir. Genel ÇATP problemleri için geliştirilen yaklaşımları uygulamak ve probleme özel yöntemlerle geliştirmek açısından, büyük ölçekli problemlerde yapılacak uygulamalar da ayrıca önem taşımaktadır. Bu problemlerde, tüm baskın nokta kümesi yerine, sadece KV'nin ilgilendiği bölgede yer alan noktaları iyi temsil eden yaklaşımlar geliştirmek amacıyla etkileşimli yöntemlerin de kullanılması önemli bir çalışma alanıdır. Ayrıca çoğu çalışmada, kriter uzayının farklı çözüm yoğunluğuna sahip farklı bölgelerinin eşit temsil edilmesi amaçlanmaktadır. Farklı özellikteki bölgelerin özelliklerini KV'nin tercihleri ile birlikte dikkate alarak yeni kalite ölçüleri geliştirmek ve bu doğrultuda yeni yaklaşımlar geliştirmek, bu konuda yararlı yeni çalışma alanlarıdır.

TEŞEKKÜR

1001-Bilimsel ve Teknolojik Araştırma Projelerini Destekleme Programı 215M844 no'lu Proje kapsamındaki desteklerinden dolayı TÜBİTAK'a teşekkür ederiz.

KAYNAKÇA

1. **Aktaş, E., Özaydın, Ö., Ülengin, F., Önsel, Ş., Ağaran, B.** 2011. "İstanbul'da İtfaiye İstasyonu Yerlerinin Seçimi İçin Yeni Bir Model," Endüstri Mühendisliği Dergisi, sayı 22 (4), s. 2-12.
2. **Armann, R.** 1987. "Solving Multiobjective Programming Problems by Discrete Representation," Optimization, vol. 20, p. 483-492.
3. **Aytuğ, H., Sayın, S.** 2009. "Using Support Vector Machines to Learn the Efficient Set in Multiple Objective Discrete Optimization," European Journal of Operational Research, vol. 193, p. 510-519.
4. **Bazgan, C., Jamain, F., Vanderpooten, D**. 2015. "Approximate Pareto Sets of Minimal Size for Multi-Objective Optimization Problems," Operation Research Letters, vol. 43, p. 1-6.
5. **Boland, N., Charkhgard, H., Savelsbergh, M.** 2015. "A Criterion Space Search Algorithm for Biobjective Integer Programming: The Balanced Box Method," INFORMS Journal on Computing, vol. 27 (4), p. 735-754.
6. **Boland, N., Charkhgard, H., Savelsbergh, M.** 2016. "The L-Shape Search Method for Triobjective Integer Programming," Mathematical Programming Computation, vol. 8 (2), p. 217-251.
7. **Burkard, R. E., Hamacher, H., Rote, G.** 1991. "Sandwich Approximation of Univariate Convex Functions with an Application to Separable Convex Programming," Naval Research Logistics, vol. 38 (6), p. 911-924.
8. **Ceyhan, G**. 2014. "Generating Representative Nondominated Point Subsets in Multiobjective Integer Programs," Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
9. **Chen, J., Li, J., Xin, B**. 2017. "DMOEA-εC: Decomposition-Based Multi-Objective Evolutionary Algorithm with the ε-Constraint Framework," IEEE Transactions on Evolutionary Computation, DOI 10.1109/TEVC.2017.2671462.
10. **Czyżżak, P., Jaszkiweicz, A.** 1998. "Pareto Simulated Annealing - a Metaheuristic Technique for Multiple-Objective Combinatorial Optimization," Journal of Multi-Criteria Decision Analysis, vol. 7, p. 34-47.
11. **Dächert, K., Klamroth, K.** 2015. "A Linear Bound on the Number of Scalarizations Needed to Solve Discrete Tricriteria Optimization Problems," Journal of Global Optimization, vol. 61 (4), p. 643-676.
12. **Deb, K.** 2001. Multiobjective Optimization Using Evolutionary Algorithms, Chichester, Wiley, U. K.
13. **Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.** 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, p. 182-197.
14. **Diakonikolas, I., Yannakakis, M.** 2009. "Small Approximate Pareto Sets for Biobjective Shortest Paths and Other Problems," SIAM Journal on Computing, vol. 39 (4), p. 1340-1371.
15. **Ehrgott, M., Gandibleux, X.** 2000. "A Survey and Annotated Bibliography of Multiobjective Combinatorial Optimization," OR Spectrum, vol. 22 (4), p. 425-460.
16. **Ehrgott, M., Gandibleux, X.** 2004. "Approximative Solution Methods for Multiobjective Combinatorial Optimization," Sociedad de Estadística E Investigación Operativa, vol. 12 (1), p. 1-89.
17. **Ehrgott, M., X. Gandibleux.** 2008. "Hybrid Metaheuristics for Multi-Objective Combinatorial Optimization," Studies in Computational Intelligence (SCI), vol. 114, p. 221-259.
18. **Eusébio, A., Figueira, J. R., Ehrgott, M.** 2014. "On Finding Representative Non-Dominated Points for Bi-Objective Integer Network Flow Problems," Computers & Operations Research, vol. 48, p. 1-10.
19. **Faulkenberg, S. L., Wiecek, M. M.** 2010. "On the Quality of Discrete Representations in Multiple Objective Programming," Optimization and Engineering, vol. 11 (3), p. 423-440.
20. **Faulkenberg, S. L., Wiecek, M. M.** 2012. "Generating Equidistant Representations in Biobjective Programming," Computational Optimization and Applications, vol. 51, p. 1173-1210.
21. **Filippi, C., Stevanato, E.** 2013. "Approximation Schemes for Bi-Objective Combinatorial Optimization and Their Application to the TSP with Profits," Computers & Operations Research, vol. 40, p. 2418-2428.
22. **Goldberg, D. E.** 1989. Genetic Algorithm in Search, Optimisation, and Machine Learning, Addison-Wesley, Boston.
23. **Goldberg, D. E., Richardson, J.** 1987. "Genetic Algorithms with Sharing for Multimodal Function Optimization," In Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, J. J. Grefenstette, Ed. Hillsdale, NJ: Lawrence, Erlbaum, 1987, p. 41-49.
24. **Hamacher, H. W., Pedersen, C. R., Ruzika, S.** 2007. "Finding Representative Systems for Discrete Bicriterion Problems," Operations Research Letters, vol. 35, p. 336-344.
25. **Ishibuchi, H., Murata, T.** 1998. "Multi-Objective Genetic Local Search Algorithm and its Application to Flowshop Scheduling," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 28 (3), p. 392-403.
26. **Jorge, J. M.** 2009. "An Algorithm for Optimizing a Linear Function over an Integer Efficient Set," European Journal of Operational Research, vol. 195, p. 98-103.

27. **Kamışlı Öztürk, Z., Kasımbeyli, N., Sağır Özdemir, M., Soyuöz Acar, M., Özçetin, E., Alegöz, M., Ceylan, G.** 2016. "Kullanıcı Tercihlerinin Dikkate Alınması Durumunda Üniversite Ders Çizelgeleme Problemi," Endüstri Mühendisliği Dergisi, sayı 27 (1), s. 2-16.
28. **Karasakal, E., Köksalan, M.** 2009. "Generating a Representative Subset of the Nondominated Frontier in Multiple Criteria Decision Making," Operations Research, vol. 57 (1), p. 187-199.
29. **Kırlık, G., Sayın, S.** 2014. "A New Algorithm for Generating all Nondominated Solutions of Multiobjective Discrete Optimization Problems," European Journal of Operational Research, vol. 232 (3), p. 479-488.
30. **Kırlık, G., Sayın, S.** 2015. "Computing the Nadir Point for Multiobjective Discrete Optimization Problems," Journal of Global Optimization, vol. 62 (1), p. 79-99.
31. **Koçanlı, M. M., Aydınbeyli, Y. E., Saraç, T.** 2012. "Eti Şirketler Grubu'nda Üretim Çizelgeleme Problemi İçin Bir Hedef Programlama Modeli ve Genetik Algoritma," Endüstri Mühendisliği Dergisi, sayı 23 (3), p. 4-21.
32. **Kohonen, T.** 1998. "The self-Organizing Map," Neurocomputing, vol. 21 (1-3), p. 1-6.
33. **Kouvelis, P., Sayın, S.** 2006. "Algorithm Robust for the Bicriteria Discrete Optimization Problem: Heuristic Variations and Computational Evidence," Annals of Operation Research, vol. 147, p. 71-85.
34. **Köksalan, M.** 1999. "A Heuristic Approach to Bicriteria Scheduling," Naval Research Logistics, vol. 46 (7), p. 777-789.
35. **Köksalan, M., Lokman, B.** 2009. "Approximating the Nondominated Frontiers of Multi-Objective Combinatorial Optimization Problems," Naval Research Logistics, vol. 56, p. 191-198.
36. **Köksalan, M., Lokman, B.** 2015. "Finding Nadir Points for Multi-Objective Integer Programs," Journal of Global Optimization, vol. 62, p. 55-77.
37. **Lokman, B., Köksalan, M.** 2013. "Finding all Nondominated Points of Multi-Objective Integer Programs," Journal of Global Optimization, vol. 57, p. 347-365.
38. **Masin, M., Bukchin, Y.** 2008. "Diversity Maximization Approach for Multiobjective Optimization," Operations Research, vol. 56 (2), p. 411-424.
39. **Mavrotas, G., Florios, K.** 2013. "An Improved Version of the Augmented e-Constraint Method (AUGMECON2) for Finding the Exact Pareto Set in Multi-Objective Integer Programming Problems," Applied Mathematics and Computation, vol. 219, p. 9652-9669.
40. **Miettinen, K., Eskelinen, P., Ruiz, F., Luque, M.** 2010. "NAUTILUS Method: An Interactive Technique in Multi-Objective Optimization Based on the Nadir Point," European Journal of Operational Research, vol. 206 (2), p. 426-434.
41. **Osman, I., Laporte, G.** 1996. "Metaheuristics: A Bibliography," Annals of Operations Research, vol. 63, p. 513-623.
42. **Özçelik, F., Saraç, T.** 2011. "Sıra Bağımlı Hazırlık Süreli İki Ölçütlü Tek Makine Çizelgeleme Problemi İçin Sezgisel Bir Çözüm Yöntemi," Endüstri Mühendisliği Dergisi, sayı 22 (4), s. 48-57.
43. **Özlen, M., Burton B. A., MacRae, C. A. G.** 2014. "Multi-Objective Integer Programming: An Improved Recursive Algorithm," Journal of Optimization Theory and Applications, vol. 160 (2), p. 470-482.
44. **Papadimitriou, C. H., Yannakakis, M.** 2000. "On the Approximability of Trade-Offs and Optimal Access of Web Sources," In: Proceedings of the 41st IEEE FOCS, IEEE Computer Society Press, p. 86-92.
45. **Ponte, A., Paquete, L., Figueira J.** 2012. "On Beam Search for Multicriteria Combinatorial Optimization Problems," In: Beldiceanu N, Jussien N, Pinson E, editors. Proceedings of the 9th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, CPAIOR 2012, May 28-June 1, 2012, Nantes, France; Lecture Notes in Computer Science, Springer, Berlin, Germany, p. 307-321.
46. **Pospelov, A.** 2009. "Approximating the Convex Edgeworth-Pareto Hull in Integer Multi-Objective Problems with Monotone Criteria," Computational Mathematics and Mathematical Physics, vol. 49 (10), p. 1686-1699.
47. **Ruhe, G., Fruhwirth, B.** 1990. "ε-Optimality for Bicriteria Programs and its Application to Minimum Cost Flows," Computing, vol. 44 (1), p. 21-34.
48. **Ruzika, S., Wiecek, M. M.** 2005. "Survey Paper: Approximation Methods in Multi-Objective Programming," Journal of Optimization Theory and Applications, vol. 126 (3), p. 473-501.
49. **Sayın, S.** 2000. "Measuring the Quality of Discrete Representations of Efficient Sets in Multiple Objective Mathematical Programming," Mathematical Programming, vol. 87 (3), p. 543-560.
50. **Sayın, S., Kouvelis, P.** 2005. "The Multi-Objective Discrete Optimization Problem: A Weighted Min-Max Two-Stage Optimization Approach and a Bicriteria Algorithm," Management Science, vol. 51 (10), p. 1572-1581.
51. **Schaffer, J. D.** 1984. "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," Ph. D. Dissertation, Vanderbilt University, Nashville, TN.
52. **Schott, J. R.** 1995. "Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization," Master's Thesis, Massachusetts Institute of Technology.
53. **Shukla, P. K., Deb, K.** 2007. "On Finding Multiple Pa-

reto-Optimal Solutions Using Classical and Evolutionary Generating Methods," European Journal of Operational Research, vol. 181, p. 1630-1652.

54. **Silverman, B. W.** 1986. Density Estimation for Statistics and Data Analysis, Chapman and Hall, London.
55. **Sipahioğlu, A., Saraç, T.** 2010. "Çok Amaçlı Sırt Çantası Probleminin Çözümüne Yeni Bir Yaklaşım: Konik Skalerleştirme," Endüstri Mühendisliği Dergisi, sayı 21 (4) s. 2-12.
56. **Srinivas, N., Deb, K.** 1995. "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Evolutionary Computation, vol. 2 (3), p. 221-248.
57. **Steiner, S., Radzik, T.** 2008. "Computing all Efficient Solutions of the Biobjective Minimum Spanning Tree Problem," Computers & Operations Research, vol. 35 (1), p. 198-211.
58. **Steuer, R. E.** 1986. Multiple Criteria Optimization: Theory, Computation, and Application, John Wiley & Sons, Inc., New York, NY.
59. **Sylva, J., Crema, A.** 2007. "A Method for Finding Well-Dispersed Subsets of Non-Dominated Vectors for Multiple Objective Mixed Integer Linear Programs," European Journal of Operational Research, vol. 180 (3), p. 1011-1027.
60. **Tuyttens, D., Teghem, J., Fortemps, P., Nieuwenhuyze, K. V.** 2000. "Performance of the Mosa Method for the Bicriteria Assignment Problem," Journal of Heuristics, vol. 6 (3), p. 295-310.
61. **Ulungu, E. L., Teghem, J.** 1995. "The Two-Phases Method: An Efficient Procedure to Solve Bi-Objective Combinatorial Optimization Problems," Foundations of Computing and Decision Sciences, vol. 20 (2), p. 149-165.
62. **Ulungu, E. L., Teghem, J., Fortemps, P. H., Tuyttens, D.** 1999. "MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems," Journal of Multi-Criteria Decision Analysis, vol. 8, p. 221-236.
63. **Van Veldhuizen, D. A.** 1999. "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," Ph. D. Thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio.
64. **Vassilvitskii, S., Yannakakis, M.** 2005. "Efficiently Computing Succinct Trade-off Curves," Theoretical Computer Science, vol. 348 (2-3), p. 334-356.
65. **Vaz, D., Paquete, L., Fonseca, C. M., Klamroth, K., Stiglmayr, M.** 2015. "Representation of the Nondominated Set in Bi-Objective Discrete Optimization," Computers & Operations Research, vol. 63, p. 172-186.
66. **Visee, M., Teghem, J., Pirlot, M., Ulungu, E. L.** 1998. "Two-Phases Method and Branch and Bound Procedures to Solve the Bi-Objective Knapsack Problem," Journal of Global Optimization, vol. 12, p.139-155.
67. **Wang, H.** 2015. "Direct Zigzag Search for Discrete Multi-Objective Optimization," Computers & Operations Research, vol. 61, p. 100-109.
68. **Wierzbicki, A. P.** 1980. "The Use of Reference Objective in Multiobjective Optimization," G. Fandel, T. Gal, eds. Multiple Criteria Decision Making, Theory and Application, Springer-Verlag, Berlin, p. 468-486.
69. **Wu, J., Azarm, S.** 2001. "Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set," Journal of Mechanical Design, Transactions of the ASME, vol. 123 (1), p. 18-25.
70. **Zhang, Q., Li, H.** 2007. "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," IEEE Transactions on Evolutionary Computation, vol. 11 (6), p. 712-731.
71. **Zhang, H., Zhou, A., Song, S., Zhang, Q., Gao, X., Zhang, J.** 2016. "A Self Organizing Multiobjective Evolutionary Algorithm," IEEE Transactions on Evolutionary Computation, vol. 20 (5), p. 792-806.
72. **Zhou, A., Qu, B.-Y., Li, H., Zhao, S-Z., Suganthan, P. N., Zhang, Q.** 2011. "Multiobjective Evolutionary Algorithms: A Survey of the State of the Art," Swarm and Evolutionary Computation, vol. 1, p. 32-49.
73. **Zitzler, E.** 1999. "Evolutionary Algorithms for Multi-Objective Optimization: Methods and Applications," Ph. D., Swiss Federal Institute of Technology, Zurich.
74. **Zitzler, E., Laumanns, M., Thiele, L.** 2002. "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," In: Giannakoglou, K., et al. (eds.) Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001), p. 95-100. International Center for Numerical Methods in Engineering, Spain.
75. **Zitzler, E., Thiele, L.** 1998. "Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study," In: Parallel Problem Solving from Nature - PPSN V, volume 1498 of the series Lecture Notes in Computer Science, Springer, Berlin, p. 292-301.
76. **Zitzler, E., Thiele, L.** 1999. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," IEEE Transactions on Evolutionary Computation, vol. 3 (4), p. 257-271.
77. **Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., Grunert da Fonseca, V.** 2003. "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," IEEE Transactions on Evolutionary Computation, vol. 7 (2), p. 117-132.