



Çok Amaçlı Tamsayı Problemlerinde Baskın Çözümler Üzerine: Analizler, Yaklaşımlar ve Uygulamalar

Program Kodu: 1001

Proje No: 215M844

Proje Yürütücüsü:
Dr. Öğr. Üyesi Banu LOKMAN

Araştırmacı:

Prof. Dr. Murat KÖKSALAN

Bursiyerler:

Gökhan ÇEYHAN

Ilgın DOĞAN

Sami Serkan ÖZARIK

DECEMBER 2018

ANKARA

ÖNSÖZ

Bu projede, çok amaçlı tamsayı problemleri (ÇATP) için internet tabanlı bir karar destek sistemi geliştirilmiştir. Bu kapsamda, ÇATP'ler için tüm baskın çözümleri etkin bir şekilde bulan web tabanlı bir algoritma tasarlanmış, bu baskın çözümlerin özellikleri ve dağılımı üzerine araştırmalar yapılmıştır. Elde edilen bilgiler ışığında, verilen herhangi bir ÇATP için baskın çözümlerin amaç fonksiyonları uzayında olası yerlerini ve dağılımını tahmin eden bir yöntem geliştirilmiştir. Bu sonuçlardan yararlanılarak, tüm baskın nokta kümesini iyi temsil edebilen altkümeler üreten algoritmalar geliştirilmiş; temsili kümenin değerlendirilmesinde kullanılacak, dağılımı da dikkate alan yeni performans ölçüleri önerilmiştir.

İki proje bursiyerinin yüksek lisans tezi çerçevesinde gerçekleştirdiği bu çalışmalardan iki yüksek lisans tezi tamamlanmıştır. Ayrıca, doktora öğrencisi olan diğer bursiyerimiz internet tabanlı bir uygulama geliştirilmesinde büyük katkı koymuş, ÇATP'ler için tüm araştırmacıların erişimine açık bir yazılım geliştirilmiştir. Proje kapsamında üretilen yüksek lisans tezleri ve internet tabanlı yazılıma ek olarak geliştirilen yöntemler ve elde edilen bulgular makaleler halinde üretilmiş ve üretilmektedir. Projenin devamında bu makalelerin uluslararası makalelerde yayınlanması hedeflenmektedir.

Tüm bu süreç kapsamında, TÜBİTAK; yüksek lisans ve doktora öğrencilerimizi desteklemiş, seyahat desteği sağlayarak uluslararası konferanslarda çalışmalarımızın sunulmasında büyük katkı sağlamıştır. Ayrıca, TÜBİTAK internet tabanlı yazılımımız için gerekli olan bilgisayar ve sunucu desteğini sağlamış, bu sayede uygulamalar ve geliştirilen araçlar tüm araştırmacıların erişimine açık olarak tasarlanmıştır. Proje ekibi, sağladığı tüm destek için TÜBİTAK'a teşekkürlerini sunar.

İçindekiler

| | |
|--|-------------|
| ÖNSÖZ | i |
| ÖZET | vii |
| ABSTRACT | viii |
| 1 GİRİŞ | 1 |
| 2 LİTERATÜR ÖZETİ | 2 |
| 2.1 Tanımlar | 2 |
| 2.2 Literatür | 4 |
| 3 GEREÇ VE YÖNTEM | 7 |
| 3.1 Çok Amaçlı Tamsayı Problemlerinde (ÇATP) Tüm Baskın Noktaların Üretilmesi . . . | 7 |
| 3.2 Çok Amaçlı Tamsayı Problemlerinde (ÇATP) Dağılıma Bağlı Temsili Kümelerin Bulunması | 13 |
| 3.2.1 Parzen Pencereleme Yaklaşımı ile Yoğunluk Tahmini | 13 |
| 3.2.2 ÇATP'lerin Dağılım Özellikleri | 14 |
| 3.2.3 L_p Yüzeyi Üzerinde PWA Yoğunluk Analizi | 16 |
| 3.2.4 Hipotetik Noktaların Yoğunluk Değerlerine Göre Kategorilere Ayrılması | 20 |
| 3.2.5 Yoğunluk Bazlı Tahmin Yöntemi (Density-based Approximation Algorithm - DAA) | 21 |
| 3.2.6 Çok Amaçlı Tamsayı Problemleri (ÇATP) için Yoğunluk Bazlı Temsilci Kümelerinin Bulunması | 23 |
| 3.2.7 Dağılım-bazlı Temsilci Bulma Algoritması (A Distribution-based Representation Algorithm - DBA) | 24 |
| 3.3 Çok Amaçlı Tamsayı Problemlerinde (ÇATP) Kullanıcı Etkileşimli Çözüm Yöntemlerinin Geliştirilmesi | 27 |
| 3.3.1 TSGA (Territory-Excluded Supported Generating Algorithm) | 27 |
| 3.3.2 TSGA-II (Territory-Excluded Supported Generating Algorithm - II) | 32 |
| 4 BULGULAR | 35 |
| 4.1 Geliştirilen Algoritmaların ve Görselleştirme Araçlarının İnternet Tabanlı Bir Uygulama Olarak Sunulması | 35 |
| 4.1.1 Uygulamalar | 35 |
| 4.1.2 Ekranlar | 37 |

| | | |
|----------|--|-----------|
| 4.2 | Geliştirilen Algoritmalar Tarafından Bulunan Çözümler için Görselleştirme Araçlarının Geliştirilmesi | 38 |
| 4.3 | DBA ile Deneyler | 46 |
| 4.4 | TSGA ve TSGA-II Yaklaşımları ile Deneyler | 48 |
| 5 | SONUÇLAR | 59 |
| 6 | KAYNAKLAR | 61 |

Şekiller

| | | |
|----|---|----|
| 1 | Alt sınır değerlerinin ağaç veri yapısı ile gösterimi | 10 |
| 2 | Yaygın olarak kullanılan çekirdek işlevleri (kernel functions) | 14 |
| 3 | Bir ÇAAP problemi için baskın nokta kümesi ve PWA yoğunluk değerleri ($N = 6573; h = 0, 2; e = 0, 05$) | 15 |
| 4 | Sırasıyla 5652 ve 8288 baskın noktaya sahip 3-amaçlı 100-parçalı ÇASP'ler için PWA Yoğunluk Değerleri ($h = 0, 2; e = 0, 05$) | 16 |
| 5 | Benzer sayıda baskın noktası olan ÇAAP ve ÇASP için PWA yoğunluk değerleri ($h = 0, 2, e = 0, 05$) | 17 |
| 6 | Baskın nokta kümesi ve hipotetik referans nokta kümesi üzerinde PWA ile hesaplanan yoğunluk değerleri ($N = 6500; h = 0, 2; e = 0, 1$) | 18 |
| 7 | Küçük ölçekli problemlerde hipotetik referans nokta kümesi üzerinde PWA ile hesaplanan yoğunluk değerleri ($h = 0, 2; e = 0, 1$) | 19 |
| 8 | Farklı büyüklükteki problemler için, hipotetik referans nokta kümesi üzerinde PWA ile hesaplanan yoğunluk değerleri ($h = 0, 2; e = 0, 1$) | 19 |
| 9 | 3-amaçlı problemler için hipotetik referans nokta kümesi üzerinde PWA ile hesaplanan yoğunluk değerleri ($h = 0, 2; e = 0, 1$) | 20 |
| 10 | 3-amaçlı 100-parçalı bir ÇASP için hipotetik noktaların PWA ile bulunan yoğunluk değerleri ve sınıflandırılması ($N = 6500; C = 3; h = 0, 2; e = 0, 1$) | 21 |
| 11 | Yoğunluk bazlı sınıflandırılan hipotetik noktaların gerçek ve tahmin edilen kategorileri ($N = 6500; C = 3; h = 0, 2; e = 0, 1$) | 23 |
| 12 | 3198 baskın noktaya sahip 3-amaçlı, 100-parçalı bir ÇASP için hipotetik noktalar arasından temsilci seçimi | 26 |
| 13 | Çözüm uzayındaki yoğunluk dağılımı ve bireysel temsil gücü ilişkisi | 29 |
| 14 | Ağırlıklı-toplam modeliyle bireysel temsil gücü yüksek noktalar üretilmesi | 29 |
| 15 | Kriter değerlerinin ağırlıklı-toplamı ve bireysel temsil gücü ilişkisi | 30 |
| 16 | nMOCO-s ve rMOCO-S çözücülerinin bulut servis olarak kullanım şeması | 36 |
| 17 | Ana sayfa görüntüsü | 38 |
| 18 | Problem dosyası yükleme ekranı | 39 |
| 19 | Çözüm sorgulama ekranı | 39 |
| 20 | Kütüphane sayfası | 39 |
| 21 | Baskın noktaların bulunduğu dosyanın yüklenmesi | 41 |
| 22 | Ana sayfa tasarımı | 42 |
| 23 | Belirli bir bölgedeki noktaların filtrelenmesi | 42 |
| 24 | Kriter uzayının interaktif bir şekilde taranması | 43 |
| 25 | Baskın noktaların z1-z2 düzlemine projeksiyonu | 43 |

| | | |
|----|--|----|
| 26 | Seçilen baskın noktaların filtreleme ile görselleştirme araçlarında karşılaştırılması . | 44 |
| 27 | Seçilen baskın noktaların dağılım grafiği üzerinde görselleştirme araçları ile karşılaştırılması | 45 |
| 28 | Seçilen baskın noktaların paralel koordinatlar grafiği üzerinde görselleştirme araçlarında karşılaştırılması | 45 |
| 29 | 3-amaçlı 100-parçalı ÇASP için baskın noktaların yoğunluk kategorileri ($N = 6500$; $C = 3$; $h = 0, 2$; $e = 0, 1$) | 46 |
| 30 | Üretilen temsilci kümesinin tüm baskın nokta kümesine göre büyüklüğü | 47 |
| 31 | Farklı boyutlardaki 3 amaçlı sırt çantası problemleri için temsilci nokta sayılarının karşılaştırılması ($\Delta = 0, 05$) | 50 |
| 32 | Farklı boyutlardaki 3 amaçlı sırt çantası problemleri için çözüm sürelerinin karşılaştırılması ($\Delta = 0, 05$) | 50 |
| 33 | 20 parçalı 3, 4 ve 5 amaçlı sırt çantası problemleri için temsilci nokta sayılarının karşılaştırılması ($\Delta = 0, 10$) | 51 |
| 34 | 20 parçalı 3, 4 ve 5 amaçlı sırt çantası problemleri için çözüm sürelerinin karşılaştırılması ($\Delta = 0, 10$) | 52 |

Tablolar

| | | |
|----|---|----|
| 1 | Test kümesi | 12 |
| 2 | Test sonuçları | 12 |
| 3 | Yoğunluk bazlı sınıflandırılan hipotetik noktaların kategorilere göre dağılımı, $C = 2$ | 20 |
| 4 | Yoğunluk bazlı sınıflandırılan hipotetik noktaların kategorilere göre dağılımı, $C = 3$ | 21 |
| 5 | DBA için sonuçlar* ($\alpha(\lambda) = 0, 10$, $\lambda = (1.00, 0.75, 0.5)$) | 47 |
| 6 | TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (3 amaçlı Sırt Çantası Problemleri deneyleri) | 53 |
| 7 | TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (4 amaçlı Sırt Çantası Problemleri deneyleri) | 54 |
| 8 | TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (5 amaçlı Sırt Çantası Problemleri deneyleri) | 55 |
| 9 | TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (3 amaçlı Atama Problemleri deneyleri) | 56 |
| 10 | TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (4 amaçlı Atama Problemleri deneyleri) | 57 |
| 11 | TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (5 amaçlı Atama Problemleri deneyleri) | 58 |

ÖZET

Günümüzde karmaşık sistemlerde, karar vericiler çoğu zaman birbiri ile çelişen çok amaçlı optimizasyon problemleri ile karşı karşıyadır. Bu problemlerde, genellikle tek bir anlamlı çözüm yoktur. Baskın çözümleri, yani en az bir amaç fonksiyonundan ödün vermeden herhangi bir amaç fonksiyonunda iyileştirme yapılması mümkün olmayan çözümleri bulmak önemlidir. Ancak, baskın çözüm sayısının problem büyüklüğü arttıkça üssel büyümesi nedeniyle; gerçek hayat problemlerinde tüm baskın çözümleri bulmak zor olduğu gibi karar verici açısından da her zaman pratik bir yaklaşım değildir.

Bu projede, Çok Amaçlı Tamsayı Problemleri (ÇATP) için farklı yaklaşımlar geliştirilmiş ve web tabanlı bir karar destek sistemi tasarlanmıştır. Bu kapsamda, ilk olarak tüm baskın çözümleri etkin bir şekilde bulan bir algoritma tasarlanmıştır. Daha sonra, yoğunluk analizinde kullanılacak bir ölçü tanımlanmış ve bu yoğunluk ölçüsü kullanılarak ÇATP'ler için ortak dağılım özellikleri elde edilmiştir. Bu özelliklerden faydalanarak, verilen herhangi bir ÇATP için baskın çözümlerin amaç fonksiyonları uzayında olası yerlerini ve dağılımını tahmin eden bir yöntem geliştirilmiştir. Bu bulgular kullanılarak, baskın nokta kümesini iyi temsil eden çözümler üreten algoritmalar geliştirilmiştir. Mevcut algoritmalarından farklı olarak, bu algoritmalar hem baskın nokta kümesinin dağılım özelliklerini hem de karar vericinin tercihlerini yansıtacak altkümeler üretecek şekilde tasarlanmıştır. Farklı ÇATP'ler üzerinde deneyler yapılarak; temsili baskın nokta kümesi üretmek için geliştirilen algoritmaların performansı değerlendirilmiş ve sonuçlarla algoritmaların iyi çalıştığı gösterilmiştir.

Proje kapsamında, internet tabanlı ve tüm araştırmacıların erişimine açık araçlar geliştirilmiştir. Bu kapsamda geliştirilen yazılım, karar vericinin belirttiği kalite ölçüleri doğrultusunda çözüm kümeleri üretebilmektedir. Ayrıca karar vericiye çözümlerin sunulmasında kullanılabilecek görselleştirme araçları web-tabanlı uygulama ile beraber tasarlanmış ve farklı ÇATP problemleri için oluşturulan test örneklerinin yayımlanacağı dijital bir test kütüphanesi de oluşturulmuştur.

Anahtar kelimeler: çok amaçlı tamsayı problemi, baskın nokta, temsili küme, etkileşimli algoritma, görselleştirme aracı.

ABSTRACT

In today's complex systems, decision makers (DMs) usually have to deal with optimization problems with conflicting objectives. In these problems, typically there does not exist a unique solution of interest. It is essential to find the nondominated points, for which an improvement in any objective is not possible without sacrificing in one of the remaining objectives. Since the number of nondominated points grows exponentially with the problem size, generating all nondominated points is not only difficult but also not practical.

In this project, we develop a web-based decision support system for Multi-objective Integer Programs (MOIPs). We first develop an exact algorithm to generate all nondominated points. In order to study the characteristics of the nondominated sets, we introduce a density measure and search for common properties of the distributions of nondominated sets. Based on this analysis, we propose a procedure that estimates possible locations and distributions of nondominated points over the criterion space. We then develop algorithms to generate a set of points that represent the nondominated set well. Different from the existing studies, the algorithms are designed to generate subsets that represent the distribution properties of the nondominated set and the preferences of the DM. We tested performance of our algorithms that generate representative subsets on different MOIPs and the results show that the algorithms work well.

We also develop an online tool that is accessible to other researchers. The software we develop is capable to interact with the DM and generate a representative set with a desired quality level. We also develop visualization tools to present the points to the DM and we present a digital library that provides a collection of test data sets for a variety of MOIPs.

Keywords: multi-objective integer program, nondominated point, representative set, interactive algorithm, visualization tool.

1 GİRİŞ

Giderek daha karmaşık hale gelen büyük organizasyonlarda üretim, çizelgeleme, yer seçimi, lojistik gibi bir çok optimizasyon probleminde karar vericiler aynı anda birden fazla kriteri dikkate alan çözümleri arar. Çoğu zaman tüm kriterleri aynı anda en iyileyen çözümün olmaması nedeni ile, bir çok karar problemi doğası gereği çok amaçlı optimizasyon problemi olarak tanımlanır ve bu problemlerde baskın çözümleri bulmak amaçlanır. Baskın noktalar, herhangi bir amaç fonksiyonundan ödün vermeden bir diğer amaç fonksiyonunda iyileşme yapmanın mümkün olmadığı olurlu çözümlerdir. Bu problemlerde, baskın noktaların sayısının problem büyüklüğü ile hızla artması aynı zamanda tamsayı karar değişkenlerinin de varlığı çözüm zorluğunu daha da artırmaktadır.

Bu projede, Çok Amaçlı Tamsayı Problemleri (ÇATP) için baskın noktalar üzerine analizler yapılmış, bu analizler doğrultusunda yaklaşımlar ve araçlar geliştirilmiştir. Bu kapsamda ilk olarak Bölüm 2'de temel tanımlar verilmiş ve ilgili literatür özetlenmiştir. Bölüm 3'te geliştirilen yaklaşımlar sunulmuştur. İlk olarak Bölüm 3.1'de tüm baskın çözümlerin bulunması için kullanılan yaklaşım sunulmaktadır. Bu yaklaşım uygulanarak, verilen bir ÇATP için tüm baskın noktalar bulunabilmekte ancak çözüm sayısının problem büyüklüğü ile hızla artması nedeniyle karar vericinin değerlendirme yapması zorlaşabilmektedir. Bu koşullarda, baskın nokta kümesini iyi temsil edebilen altkümeler bulmak hedeflenir. Mevcut yaklaşımlar, bu temsili kümeyi üretirken farklı kalite ölçüleri kullanmakta ancak bu değerlendirmede baskın nokta kümesinin çözüm uzayına dağılımını dikkate almamaktadır. Bu kapsamda, Bölüm 3.2'de önce ÇATP'lerin baskın nokta kümelerinin dağılımı üzerine detaylı analizler yapılmıştır. Burda edinilen bilgiler ışığında, verilen herhangi bir ÇATP için baskın noktaların olası yerlerini ve yoğunluk seviyelerini tahmin edebilen bir yöntem geliştirilmiştir. Daha sonra, bu yöntem kullanılarak gerçek temsili baskın nokta kümeleri üreten bir algoritma tasarlanmıştır. Karar vericinin belirlediği bir kalite seviyesini garantilemek ve karar vericiyi de çözüm sürecine dahil etmek amacıyla Bölüm 3.3'te iki algoritma geliştirilmiştir. Bu algoritmalar, ÇATP için baskın nokta kümesinin tipik özelliklerinden faydalanarak amaç fonksiyonlarını ağırlıklandırmakta ve bu sayede her adımda daha önceki çözümler tarafından baskılanmayan ya da temsil edilemeyen yeni baskın noktaları bulmaktadır.

Bölüm 3'te önerilen yaklaşımlara ait yazılımlar, araçlar ve uygulamalar bulgularıyla beraber Bölüm 4'te sunulmaktadır. İlk olarak Bölüm 4.1'de geliştirilen algoritmaların ve görselleştirme araçlarının internet tabanlı bir uygulama olarak sunulması için yapılan çalışmalar ve araçlarımız yer almaktadır. Tüm baskın nokta kümesinin veya temsili altkümenin bulunması için tasarlanan bu uygulama, hem orta ölçekli problemlerin çözülmesi hem de sezgisel yöntemlerin performansını değerlendirilmesi amacıyla kullanılabilir. Bu kapsamda, ayrıca tüm araştırmacıların erişimine açık bir dijital kütüphane de tasarlanmıştır. Daha sonra, Bölüm 4.2'de üretilen baskın noktaların karar vericiye sunulması için geliştirilen görselleştirme araçları sunulmaktadır. Bölüm 4.3 ve 4.4'te ise temsili baskın nokta kümesi bulmak için geliştirilen yeni yaklaşımların performansı değerlendirilmekte ve deney bulguları sunulmaktadır. Son olarak, Bölüm 5'te elde edilen sonuçlar tartışılmaktadır.

2 LİTERATÜR ÖZETİ

ÇATP'lerin, gerçek hayatta bir çok uygulaması olmakla birlikte çözüm zorluğu da amaç fonksiyonu sayısına ve problem büyüklüğüne bağlı olarak hızla artmaktadır. Bu nedenle son dönemde, ÇATP'ler için bir çok yaklaşım önerilmiştir. Bu kapsamda, ilk olarak gerekli tanımlar verilmiş ve daha sonra ilgili çalışmalar tartışılmıştır.

2.1 Tanımlar

Bir ÇATP problemi aşağıdaki şekilde yazılabilir:

$$\text{"Maks"}_{\mathbf{x} \in \mathbf{X}} z(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \dots, z_m(\mathbf{x})) \quad (1)$$

Burada, $z_i(\mathbf{x})$ i . amaç fonksiyonunu, m amaç fonksiyonu sayısını, \mathbf{x} tamsayı karar vektörünü ve \mathbf{X} olurlu çözüm (karar) uzayını göstermektedir. \mathbf{X} uzayına karşılık gelen amaç fonksiyonu uzayı ise $\mathbf{Z} = \{\mathbf{z} | \mathbf{z} = z(\mathbf{x}), \mathbf{x} \in \mathbf{X}\}$ olarak tanımlanmaktadır.

Tanım 1 (Etkin çözüm). Eğer bir $\mathbf{x} \in \mathbf{X}$ olurlu çözümü için $z_i(\mathbf{x}') \geq z_i(\mathbf{x})$ $i = 1, \dots, m$ koşullarını ve en az bir amaç fonksiyonunda $z_i(\mathbf{x}') > z_i(\mathbf{x})$ koşulunu sağlayan olurlu bir $\mathbf{x}' \in \mathbf{X}$ çözümü var ise \mathbf{x}' çözümünün \mathbf{x} çözümünü baskıladığı söylenir. \mathbf{x} çözümünü baskılayan olurlu bir \mathbf{x}' çözümü yok ise $\mathbf{x} \in \mathbf{X}$ çözümü etkin çözümdür.

Etkin olan çözümlerin kümesi \mathbf{X}_E olarak tanımlanmıştır.

Tanım 2 (Baskın nokta). Eğer $\mathbf{x} \in \mathbf{X}$ çözümü etkin ise, $\mathbf{x} \in \mathbf{X}_E$, ona karşılık gelen amaç fonksiyonu vektörü $z(\mathbf{x})$ baskın noktadır.

$\mathbf{Z}^B = \{\mathbf{z} | \mathbf{z} = z(\mathbf{x}), \mathbf{x} \in \mathbf{X}_E\}$ tüm baskın nokta kümesini, $N = |\mathbf{Z}^B|$ ise toplam baskın nokta sayısını temsil etmektedir.

Tanım 3 (Zayıf Baskın Nokta). Eğer $\mathbf{x}, \mathbf{x}' \in \mathbf{X}$ olurlu çözümleri için $z_i(\mathbf{x}') > z_i(\mathbf{x})$ $i = 1, \dots, m$ koşulu sağlanıyorsa \mathbf{x}' çözümünün \mathbf{x} çözümünü tam baskıladığı söylenir. Eğer böyle olurlu bir $\mathbf{x}' \in \mathbf{X}$ çözümü yok ise $\mathbf{x} \in \mathbf{X}$ çözümüne zayıf etkin çözüm ve $z(\mathbf{x})$ noktasına zayıf baskın nokta denir.

Zayıf baskın noktalar içinde tüm baskın noktalar bulunmaktadır, ancak zayıf baskın olan ama baskın olmayan noktalar da vardır.

Tanım 4 (Destekli Baskın Nokta). Eğer bir baskın nokta, $\mathbf{z}^* \in \mathbf{Z}^B$, için $\mathbf{z}^* = \operatorname{argmax}_{\mathbf{z} \in \mathbf{Z}} \sum_{i=1}^m w_i z_i$ ve $w_i > 0$ koşullarını sağlayan bir ağırlık vektörü $\mathbf{w} = (w_1, \dots, w_m)$ bulunabiliyorsa \mathbf{z}^* noktası destekli baskın noktadır.

Tanım 4'te gösterildiği gibi amaç fonksiyonlarının ağırlıklandırılmasıyla oluşan yeni amaç fonksiyonu kullanılarak ÇATP'ler tek amaçlı optimizasyon problemine dönüştürülebilir. Bu şekilde, baskın noktalar ancak pozitif ağırlıklar kullanılırsa bulunabilir. Herhangi bir kriterin ağırlığı sıfır olduğunda diğerlerinin hepsi pozitif olsa bile zayıf baskın olan ama baskın olmayan nok-

olarak bulunabilir. Bu nedenle, tüm amaç fonksiyonlarının ağırlıklarının sıfırdan büyük seçilmesi önemlidir. Bu şekilde bulunabilen destekli baskın noktalara ek olarak; baskın olan ancak destekli olmayan baskın noktalar da vardır. Bu noktaların bu yöntem ile bulunması mümkün değildir ve desteksiz noktaların sayısı Lokman (2017) tarafından da gösterildiği gibi problem büyüklüğü arttıkça destekli noktalara göre çok daha büyük bir hızla artmaktadır.

Tanım 5 (İdeal Noktası). İdeal noktası, $\mathbf{z}^{IP} = (z_1^{IP}, z_2^{IP}, \dots, z_m^{IP})$, her bir amaç fonksiyonunun alabileceği en iyi değerlerden oluşur. Maksimizasyon problemlerinde $z_i^{IP} = \max_{\mathbf{x} \in \mathbf{X}}(z_i(\mathbf{x}))$ $i = 1, \dots, m$ olarak, minimizasyon problemlerinde $z_i^{IP} = \min_{\mathbf{x} \in \mathbf{X}}(z_i(\mathbf{x}))$ $i = 1, \dots, m$ olarak tanımlanır.

Tanım 6 (Nadir Noktası). Nadir noktası, $\mathbf{z}^{NP} = (z_1^{NP}, z_2^{NP}, \dots, z_m^{NP})$, her bir amaç fonksiyonunun etkin çözümler içerisinde alabileceği en kötü değerlerden oluşur. Maksimizasyon problemlerinde $z_i^{NP} = \min_{\mathbf{x} \in \mathbf{X}_E}(z_i(\mathbf{x}))$ $i = 1, \dots, m$ olarak, minimizasyon problemlerinde ise $z_i^{NP} = \max_{\mathbf{x} \in \mathbf{X}_E}(z_i(\mathbf{x}))$ $i = 1, \dots, m$ olarak tanımlanır.

İdeal ve nadir noktaları, amaç fonksiyonlarının ölçeklendirilmesinde kullanıldığı için algoritmalar için önem taşır. İdeal noktası, her bir amaç fonksiyonunun ayrı ayrı en iyilenmesi ile yani m tane tek amaçlı model çözerek bulunabilir. Nadir noktası, amaç fonksiyonlarının tüm olurlu çözümler içerisinde değil sadece etkin çözümler içinde aldıkları en kötü değerlerden oluşur. Bu nedenle bulması daha zordur. Ancak, son dönemde, ÇATP için nadir noktasını bulan etkin algoritmalar geliştirilmiştir (bakınız Jorge, 2009; Köksalan ve Lokman, 2015; Kirlik ve Sayın, 2015).

Tüm baskın noktaları üretmek hem maliyetli hem de pratik olmayan bir yaklaşımdır. Bu nedenle baskın nokta kümesini belirli kalite ölçülerine göre iyi temsil edebilen altkümeler, $\mathbf{R} \subseteq \mathbf{Z}^B$, bulabilmek önemlidir.

Sayın (2000), temsilci kümelerin, kalitesini değerlendirebilmek için aşağıdaki kalite ölçülerini önermiştir. Bu tanımlarda, $d(\cdot, \cdot)$, \mathbb{R}^m uzayında tanımlı uzaklık ölçüsünü göstermektedir.

Tanım 7 (Kapsama Ölçüsü): Her bir baskın nokta $\mathbf{z} \in \mathbf{Z}^B$ için, $\mathbf{R} \subseteq \mathbf{Z}^B$ temsilci kümesi içinde, $d(\mathbf{y}, \mathbf{z}) \leq \alpha$ koşullarını sağlayan bir temsilci $\mathbf{y} \in \mathbf{R}$ var ise, \mathbf{R} kümesine \mathbf{Z}^B kümesinin d_α – *temsili* denir. Tüm baskın noktalar için bu koşulu sağlayan en küçük α değerine \mathbf{R} kümesinin \mathbf{Z}^B kümesini kapsama ölçüsü denir.

Tanım 7’de verilen ölçü, üretilen bir temsili kümenin bir problemdeki bütün baskın noktaları en az hata ile kapsamaları için kullanılmaktadır. Burada; her bir baskın nokta için temsili kümedeki en yakın nokta, temsilci olarak atanır ve yapılan en büyük kapsama hatası kapsama ölçüsünü verir. Öyle ki; Tchebycheff temelli bir uzaklık ölçüsü kullanıldığında, kapsama ölçüsü alttaki gibi hesaplanabilir:

$$\alpha = \max_{\mathbf{z} \in \mathbf{Z}^B} (\min_{\mathbf{y} \in \mathbf{R}} (\max_{1 \leq i \leq m} (z_i - y_i))) \quad (2)$$

Maksimizasyon problemi için Denklem 2’de tanımlanan bu ölçüde, temsil edilen noktanın temsilcisinden daha iyi olduğu kriterler kapsama hatasına pozitif katkı yapmaktadır.

Tanım 8 (Eşit Dağılım): Verilen $R \subseteq Z^B$ temsili kümesi için eşit dağılım ölçüsü

$$\delta = \min_{\substack{y^1, y^2 \in R \\ y^1 \neq y^2}} d(y^1, y^2) \quad (3)$$

olarak tanımlanmıştır.

Tanım 8’de verilen eşit dağılım ölçüsü, üretilen temsili baskın noktaların amaç fonksiyonu uzayında eşit dağılmasını hedeflemektedir. Bu nedenle, en yakın iki temsilci arasındaki uzaklık eşit dağılım ölçüsü olarak tanımlanır ve bu uzaklığın daha büyük olduğu temsilci kümeleri bulmak hedeflenir.

Tanım 9 (Küme Niceliği): $R \subseteq Z^B$ kümesinin büyüklüğü, $|R|$, küme niceliği olarak tanımlanmıştır.

Burada, hesaplama maliyeti de dikkate alındığından, amaç baskın nokta kümesinin en az nokta ile temsil edebilmektir. Bu kalite ölçüleri birbiri ile ilişkilidir. Kalite ölçülerinin herhangi birinde yapılan iyileştirme, diğer kalite ölçülerini iyi veya kötü yönde etkileyebilir. Örneğin, temsilci kümesindeki nokta sayısının artırılması, çözüm zorluğunun artmasına neden olurken, kapsama hatasının azalmasına ve dolayısıyla noktaların temsil edilme kalitesinin artmasına neden olabilir. Faulkenberg ve Wiecek (2010), çok amaçlı optimizasyon problemleri için kalite ölçülerini Sayın (2000)’ın önerdiği gibi üç sınıfta incelemektedir. Lokman (2017), bu proje kapsamında ÇATP’ler için kullanılan farklı kalite ölçüleri ve temsili küme bulan yaklaşımlar üzerine kapsamlı bir inceleme çalışması sunmaktadır.

2.2 Literatür

ÇATP’lerde, birden fazla amaç fonksiyonu olması ve doğası gereği bu amaç fonksiyonlarının birbiriyle çelişmesi nedeniyle tek bir etkin çözüm yoktur. Bu nedenle tüm etkin çözümleri, ya da baskın noktaları, bulmak önemlidir. Bu alandaki öncü çalışmalar sadece iki amaçlı problemleri kapsamaktadır ve literatürde birçok örneği mevcuttur. Problem özelinde geliştirilen algoritmaların yanında, epsilon-kısıtı yöntemi ve iki-fazlı algoritma iki amaçlı tamsayı problemlerinin çözümü için sıklıkla kullanılan yöntemlerdir (bakınız Haimes, 1971; Przybylski vd., 2008). Bu yöntemlerde, kriterlerden biri amaç fonksiyonu olarak kullanılırken, maksimizasyon (minimizasyon) problemlerinde diğer kriterler ile alt (üst) sınır koyan kısıtlar eklenmektedir. Bu kısıtlar sistematik olarak değiştirildiğinde etkin çözümler bulunabilir. Epsilon kısıtı yönteminde, diğer kısıtın amaç fonksiyonunda bulunmaması nedeni ile zayıf baskın ama baskın olmayan çözümler bulunabilir. Bu nedenle, baskın nokta bulmayı garantilemek için iki farklı yöntem önerilmiştir. İlkinde, diğer kriterler amaç fonksiyonunda seçilen kriterin yanında çok küçük pozitif bir sayı ile çarpılarak eklenir. Böylece yine diğer kriterler için belirlenen sınırlar içinde seçilen kriterin eniyilendiği çözüm bulunur. Hem de bulunan noktanın baskın nokta olması garantilenir. Bu yöntem, modifiye edilmiş epsilon kısıtı yöntemidir. İki-fazlı algoritmalarda ise ilk fazda, seçilen kriter belirlenen sınırlar içinde eniyilenir ve ikinci fazda bu kriter en iyi değerinde sabitlenerek diğer kriterlerin toplamı eniyilenerek baskın nokta bulması garantilenir. İki kriterli problemlerde, bir kriterin amaç fonksiy-

onu olarak kullanılırken diğer kriter için tek kısıt eklenmesi nedeni ile bu yöntemlerin uygulanması daha kolaydır. Ancak kriter sayısı arttıkça uygulamada zorluklar ortaya çıkar.

İkiden fazla amaçlı problemlere yönelik çözüm yöntemleri ise son on yılda ortaya çıkmıştır. Bu çalışmalardan bazıları sadece kısıtlı sayıda amaç fonksiyonu için geçerli olan veya belirli bir problem tipine yöneliktir. Çok sayıda amaç fonksiyonuna sahip genel bir tamsayı problemi çözme kabiliyetine sahip yöntemler çok kısıtlıdır. Bu yöntemlerden ilki Sylva ve Crema (2004) tarafından önerilen ve elde edilen her baskın çözüm için baz modele yeni ikili değişken ve kısıtların eklenmesi ile kısıtlanan uzayın ayrıştırılmasına dayanan bir algoritmadır. Eklenen ikili değişken ve kısıtlardan dolayı model çok hızlı şekilde çözülemeyecek boyutlara ulaştığı için pratik değildir. Diğer çözüm yöntemleri ise (bakınız Laumanns vd., 2006; Özlen ve Azizoğlu, 2009; Przybylski vd., 2010; Dhaenens vd., 2010; Lokman ve Köksalan, 2013; Özlen vd. 2014; Kirlik ve Sayın, 2014) amaç fonksiyonu uzayının sistematik bir şekilde parçalanıp baskın çözümlerin olası bölgelerde aranmasına dayanmaktadır. Bu yöntemler genellikle üç amaçlı problemler üzerinde test edilmiş olup Lokman ve Köksalan (2013) ve Özlen vd. (2014) dört amaca kadar, Kirlik ve Sayın (2014) ise beş amaca kadar olan test örnekleri üzerinde deney yapmış ve raporlamışlardır. Bu yöntemler arasında, Lokman ve Köksalan (2013) ve Kirlik ve Sayın (2014) mevcut literatürdeki ilk etkin çözüm yöntemleri olarak ön plana çıkmaktadır. Daha sonra, Boland vd. (2016), Dächert ve Klamroth (2015), Klamroth vd. (2015) ve Dächert vd. (2017) amaç fonksiyonunun daha etkin bir şekilde bölünmesi üzerine çalışmalar sunmuştur.

ÇATP'ler son zamanlarda birçok araştırmacı tarafından ilgi görmüştür ve yeni araştırmalara son derece açık bir alandır. Ancak çözüm yaklaşımının etkinliğinden bağımsız olarak, tüm baskın çözüm kümesini bulmak çok sayıda baskın çözüme sahip problemler için pratik bir yöntem olmaktan uzaktadır. Harcanan hesaplama maliyeti çok hızlı artmakta ve elde edilen çözüm kümesi karar verici tarafından pratik olarak analiz edilemeyecek kadar büyümektedir. Bu nedenle, ÇATP'ler için baskın nokta kümesinin temsili alt kümelerini bulma üzerine araştırmalar yapılmaktadır. Sylva ve Crema (2007), Sylva ve Crema(2004) tarafından tüm baskın noktaları bulmak için geliştirilen algoritmayı, her adımda baskılanan uzaydan Tchebycheff uzaklığına göre en uzak baskın noktayı bulacak şekilde değiştirmişlerdir. Böylece, her adımda amaç fonksiyonu uzayına iyi dağılmış bir baskın çözüm kümesi elde etmeyi amaçlamışlardır.

Masin ve Bukchin (2008) ise temsili alt kümelerin değerlendirilmesi için yeni bir kalite ölçüsü önermiş ve her adımda bu ölçüyü en azlayacak baskın noktayı aramışlardır. Algoritma sonunda, üretilmeyen her baskın nokta için onu temsil eden ve herhangi bir amaçta en fazla belirlenmiş olan kalite ölçüsü kadar kötü olan bir temsili baskın noktanın olmasını garanti etmektedir. Ancak Sylva ve Crema (2007)'nin yönteminde olduğu gibi bu yöntemde de çözülmesi gereken matematiksel model ve hesaplama maliyeti hızla büyümektedir.

Ceyhan vd. (2019), Masin ve Bukchin (2008) tarafından önerilen kalite ölçüsünü baz alarak, daha etkin çözüm yöntemleri geliştirmiştir. Bu yöntemlerden ilkinde, Masin ve Bukchin (2008) tarafından önerilen yöntem farklı bir arama algoritması ile uygulanarak çözüm süreleri üzerinde önemli iyileştirmeler yapılmıştır. Geliştirilen ikinci yöntem, karar verici tarafından bildirilecek olan kalite ölçüsü eşik değerini garanti etmekte ve ilk algoritmaya göre daha hızlı çalışmaktadır. Diğer bir yöntemde ise, daha kaliteli bir temsili küme bulmak amaçlanmıştır. Baskın noktaların bulun-

duđu yerlere bir yüzey yardımı ile yaklaşılarak, bu yüzey üzerinde ve bu yüzeyi iyi temsil eden belirli sayıda temsili nokta alınıp bu noktalara yakın baskın çözümler üretilmiştir.

Etkin çözümleri bulmanın güçlüğüne bağılı olarak, Ehrgott ve Gandibleux (2000) tarafından da belirtildiğı gibi; etkin çözümleri bulmak için yaklaştırma yöntemleri ve sezgisel yaklaşımlar geliştirilmiştir. Köksalan (1999) etkin çözümlerin bulunduğu bölgeleri yüzeyler vasıtasıyla yaklaşık olarak tanımlayan bir yöntem geliştirmiş, Köksalan ve Lokman (2009) bu yöntemi ikiden fazla amaçlı bileşi problemleri için genelleştirmiştir. Üç ve dört amaçlı sırt çantası, en kısa patika ve kapsayan ağaç problemleri üzerinde deneyler yapmışlardır. Lokman ve Köksalan (2014) ise ÇATP için etkin çözümlerin yer aldığı bölgeyi yaklaşık olarak tanımlayan bir yüzey kullanarak karar vericinin tercih ettiğı çözümleri bulan etkileşimli bir algoritma geliştirmiştir. Phelps ve Köksalan (2003) çok amaçlı bileşi problemleri için etkileşimli evrimsel bir sezgisel yöntem geliştirmiştir. İki, üç ve dört amaçlı sırt çantası ve kapsayan ağaç problemlerinde deneyler yapmışlardır. Çok amaçlı bileşi problemlerinin çözüm güçlüğü nedeni ile karar verici ile etkileşim kuran bu algoritma tercih edilen bölgelere odaklanmaktadır. Zitzler ve Thiele (1999), çok amaçlı bileşi problemlerinin tüm etkin çözümlerini bulmak için evrimsel bir yöntem önermiş ve bu yöntemi bazı evrimsel algoritmalarla karşılaştırmıştır. Zitzler ve Thiele (1999), iki, üç ve dört amaçlı sırt çantası problemleri üzerinde deneyler yapmıştır. Lokman vd. (2016), karar vericinin tercihlerinin kuasikonveks bir değer fonksiyonunu ile uyumlu olduğunu varsayarak, ÇATP'ler için en çok tercih edilen çözümü bulan bir algoritma tasarlamıştır. Lokman vd. (2018) ise en çok tercih edilen çözümü yaklaşık olarak bulan etkin bir algoritma geliştirmiştir.

ÇATP'ler için temsili nokta kümesi bulmayı amaçlayan mevcut yaklaşımlar, baskın nokta kümesinin amaç uzayındaki dağılımını dikkate almamaktadır (bakınız Lokman, 2017). Ancak, bir baskın çözüm alt kümesinin, iyi bir temsili alt küme olabilmesi için amaç fonksiyonu uzayındaki dağılımının baskın çözüm kümesininkine benzer olması istenebilir. Bu proje kapsamında, ilk olarak Lokman ve Köksalan (2013) tarafından önerilen algoritma geliştirilerek tüm baskın noktaları bulan bir algoritma tasarlanmış, Bölüm 3.1'de sunulmuştur. Daha sonra, elde ettiğimiz bu çözümlerin dağılım özellikleri üzerine analizler yapılmış, bu analizler sonucunda baskın çözüm kümesini temsil eden altkümeler üreten yöntemler geliştirilmiştir. Bu yöntemler, mevcut literatürden farklı olarak baskın nokta kümesinin temsilinde amaç fonksiyonu uzayındaki dağılımı da dikkate almaktadır. Bölüm 3.2'de sunulan yaklaşım, ilk olarak verilen bir ÇATP için baskın noktaların amaç kriteri uzayında yerlerini ve yoğunluğunu tahmin etmektedir. Daha sonra, bu dağılıma bağılı olarak bölgelerden temsilci baskın noktalar üretmektedir. Bölüm 3.3'te sunulan yaklaşım ise, baskın nokta kümesinin dağılım özelliklerinden yola çıkarak kriterler için ağırlıklar atamakta, bu ağırlıkları kullanarak belirli bir kalite ölçüsünü garantileyen temsilci bir baskın nokta kümesi üretmektedir. Bölüm 3'te sunulan yaklaşımlar kullanılarak geliştirilen araçlar ve yapılan deneyler Bölüm 4'te yer almaktadır. Bu kapsamda ilk olarak Bölüm 4.1'de, geliştirilen algoritmaların ve görselleştirme araçlarının internet tabanlı bir uygulama olarak sunulması için geliştirilen yazılımlar sunulmaktadır. Daha sonra, Bölüm 4.2'de geliştirilen görselleştirme araçları yer almaktadır. Bölüm 3.2'de ve 3.3'te geliştirilen yeni algoritmalar için yapılan deney sonuçları sırasıyla Bölüm 4.3'te ve 4.4'te sunulmaktadır. Son olarak, Bölüm 5'te sonuçlar tartışılmaktadır.

3 GEREÇ VE YÖNTEM

3.1 Çok Amaçlı Tamsayı Problemlerinde (ÇATP) Tüm Baskın Noktaların Üretilmesi

Bu bölümde, ÇATP'lerde tüm baskın noktaların üretilmesi için daha az hesaplama maliyetli yöntemlerin geliştirilmesi konusunda çalışılmıştır. ÇATP'ler için tüm baskın nokta kümesini elde etmek, tüm baskın noktaların temsili bir alt kümesini bulmayı veya bu çözümlere amaç fonksiyonu uzayında yaklaşmayı hedefleyen yöntemlerin performansını değerlendirmek için gereklidir. Ayrıca, tüm baskın noktaların üretilmesi, baskın nokta sayısının çok fazla olmadığı tahmin edilen problemlerde ve yeterli işlemci veya işlem süresi bütçesi olan uygulamalarda tercih edilebilmektedir. Bu amaç doğrultusunda geliştirmiş olduğumuz yöntem, verilen bir ÇATP için tüm baskın çözümleri bulmayı garanti etmektedir. Yöntem her döngüde yeni bir baskın nokta bulur ve bir sonraki baskın noktanın bulunduğu arama uzayını daraltarak ilerler. Her döngüdeki arama uzayı, olurlu amaç fonksiyonu uzayından elde edilen baskın noktalar tarafından baskılanan uzayın çıkarılması ile elde edilen kümenin bir alt kümesidir. Geliştirdiğimiz yöntem, bu arama uzaylarının etkin bir şekilde taranmasını sağlayarak, çözülecek model sayısını azaltmaktadır. Buna ek olarak, çok boyutlu amaç fonksiyonu uzayındaki arama alt kümelerinin üretilmesi, saklanması ve güncellenmesi fonksiyonlarının verimli bir şekilde ve herhangi bir sayıda amaç fonksiyonu için geçerli olacak şekilde yerine getirilmesini sağlayacak veri yapısı geliştirilmiştir.

Bildiğimiz kadarı ile, ilk olarak Sylva ve Crema (2004) verilen bir ÇATP için tüm baskın noktaları bulan bir yöntem önermiştir. Bu yöntemde, her adımda bir baskın nokta üretilip; bu nokta tarafından baskılanan uzay, arama uzayından çıkarılmaktadır. Sylva ve Crema (2004), her adımda aşağıdaki modeli çözmeyi önermişlerdir:

(SC):

$$\text{Maks } \sum_{i=1}^m w_i z_i(\mathbf{x})$$

Kısıtlar

$$z_i(\mathbf{x}) \geq z_i^j(\mathbf{x}) + \rho t_{ij} - M(1 - t_{ij}) \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, n$$

$$\sum_{i=1}^m t_{ij} \geq 1 \quad \forall j = 1, \dots, n$$

$$t_{ij} \in \{0, 1\} \quad \forall i = 1, \dots, m, \quad \forall j = 1, \dots, n$$

$$\mathbf{x} \in \mathbf{X}$$

(SC) modelinde, m amaç fonksiyonu sayısını göstermekte olup, üretilen baskın noktalar için j , amaç fonksiyonları için ise i indisi kullanılmıştır. \mathbf{X} kümesi problemin olurlu kümesini, $z_i(\mathbf{x})$ ise \mathbf{x} olurlu çözümünün i amaç fonksiyonundaki değerini göstermektedir. İkili değişkenler olan t_{ij} değişkeni üzerindeki kısıt yardımıyla üretilen çözümün mevcut adıma kadar üretilmiş her bir j baskın noktasından en az bir i amaç fonksiyonunda daha iyi olması sağlanarak (yeterince küçük pozitif bir ρ değeri kullanılarak) yeni bir çözüm bulunması garanti edilmektedir. M ise t_{ij} ikili değişkeni sıfır değerini aldığı anda ilgili kısıtı etkisiz yapacak kadar büyük bir sayı olarak tanımlanmıştır. Amaç fonksiyonundaki w_i 'ler baskın noktayı garanti etmek için sıfırdan büyük

değer almalıdır. Bu yöntem ile $(n+1)$ inci baskın çözümü bulmak için çözülecek (SC) modelinde ek olarak $n.m$ sayıda ikili değişken ile $n(m+1)$ tane kısıt gerekmektedir. Çok sayıda baskın noktaya sahip problemler için (SC)'nin çözülmesi pratik olmadığından mevcut çalışmalar arama uzayını alt uzaylara ayırıştırarak bu alt uzayları sistematik bir şekilde aramaya dayalı yöntemler geliştirmişlerdir (bakınız Laumanns vd., 2006; Özlen ve Azizoğlu, 2009; Przybylski vd., 2010; Dhaenens vd., 2010; Lokman ve Köksalan, 2013; Özlen vd. 2014; Kirlik ve Sayın, 2014; Dächert ve Klamroth, 2015; Boland vd., 2016). Geliştirilen son yöntemler oldukça rekabetçi sonuçlar ortaya koymaktadır. Lokman ve Köksalan (2013), Özlen vd. (2014) ve Kirlik ve Sayın (2014) çalışmalarında geliştirilen yöntemler her sayıda amaç fonksiyonu için çalışabilmektedir. Dächert ve Klamroth (2015) ve Boland vd. (2016) tarafından geliştirilen algoritmalar ise sadece üç amaçlı problemler için geçerlidir.

(SC) modelinin dezavantajı, her bir yeni baskın çözüm üretmek için bu baskın çözümün bulunduğu uzayı ayırt edebilmek için daha karmaşık bir problemin çözülmek zorunda olmasıdır. Bu şekilde bir çözüm yaklaşımı yerine, baskın noktaların bulunduğu arama uzayı sistematik bir biçimde alt uzaylara bölmek ve bu alt uzaylarda aynı zorluktaki problemleri çözerek baskın noktaları üretmek mümkündür. Aşağıdaki kısımda, bu fikir üzerine kurulu çözüm yaklaşımımız anlatılmaktadır. İlk olarak Lokman ve Köksalan (2013) tarafından önerilen algoritma özel ağaç veri yapısı ile geliştirilerek, herhangi sayıda amaç fonksiyonu altında çalışabilecek hale getirilmiştir.

\mathbf{Z} kümesi olurlu çözüm uzayını, $\mathbf{R} = \{\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^n\}$ kümesi de üretilmiş olan baskın noktaları gösterebilir. Boş bir \mathbf{R} kümesi ile başlayıp, kümeyi her döngü sonucunda bulunan baskın noktayı ekleyerek genişletelim. \mathbf{z}^j noktası tarafından baskılanan olurlu amaç fonksiyonu uzayını \mathbf{z}_D^j , baskılanan olurlu amaç fonksiyonu uzayını ise \mathbf{Z}_D ile gösterirsek:

- $\mathbf{z}_D^j = \{\mathbf{z} \in \mathbf{Z} \mid z_i \leq z_i^j, i = 1, \dots, m\}$
- $\mathbf{Z}_D = \cup_{j=1}^n \mathbf{z}_D^j$

olarak tanımlanır.

Bu durumda, $\mathbf{S} = \mathbf{Z} \setminus \mathbf{Z}_D$ kümesi \mathbf{R} kümesi dışındaki baskın noktaların bulunduğu olurlu amaç fonksiyonu kümesini gösterir. Tek amaçlı problemlerde, $m = 1$ ve $n \leq 1$ olduğu için \mathbf{S} boş kümedir. Çok amaçlı problemlerde, $m \geq 2$ ve $n \geq 1$, bir sonraki baskın nokta, $\mathbf{z} = (z_1, z_2, \dots, z_m)$, eğer var ise, aşağıda verilen $\mathbf{w} > 0$ ağırlık vektörü ile problem çözülerek bulunabilir:

$$(P_S) : \text{Maks}_{\mathbf{z} \in \mathbf{S}} \mathbf{wz}$$

Bununla birlikte, \mathbf{Z} kümesinin kısıtlı ("bounded") olduğu durumda, \mathbf{S} kümesi sonlu sayıdaki çökyüzlü ("polyhedral") kümeler içinde kalan olurlu amaç fonksiyonu değer kümelerinin birleşimi şeklinde yazılabilmektedir:

$$\mathbf{S}_k = \{\mathbf{z} \in \mathbf{Z} \mid z_i \geq b_i^k, \quad i = 1, \dots, m, \quad \mathbf{k} \in \mathbf{K} \quad \mathbf{b}^k \in \mathbb{R}^m\}, \quad \mathbf{S} = \cup_{\mathbf{k} \in \mathbf{K}} \mathbf{S}_k.$$

Eğer $\bar{\mathbf{z}} \in \mathbf{S}_{\bar{\mathbf{k}}}$, o zaman $\bar{z}_i \geq b_i^{\bar{\mathbf{k}}} - u_i^{\bar{\mathbf{k}}}$, $\mathbf{k} \in \mathbf{K}$, $\mathbf{k} \neq \bar{\mathbf{k}}$ kısıtlarını sağlayan $u_i^{\bar{\mathbf{k}}} \geq 0$ değerleri vardır. Bu durumda, (P_S) aşağıdaki gibi bir tamsayı programı ile çözülebilir:

(P_S):

Maks wz

Kısıtlar

$$z_i \geq b_i^k - w_i^k(1 - y^k) \quad k \in \mathbf{K} \quad i = 1, \dots, m$$

$$\sum_{k \in \mathbf{K}} y^k = 1$$

$$\mathbf{z} \in \mathbf{Z}$$

Geliştirdiğimiz yöntemdeki temel fikir, (P_S) problemini çözmek yerine bu problemi alt problemlere, (P_{S_k}), bölerek bulunan \mathbf{z} noktalarından en yüksek wz değerine sahip olanı bir sonraki baskın çözüm olarak \mathbf{R} kümesine eklemektir.

(P_{S_k}):

Maks wz

Kısıtlar

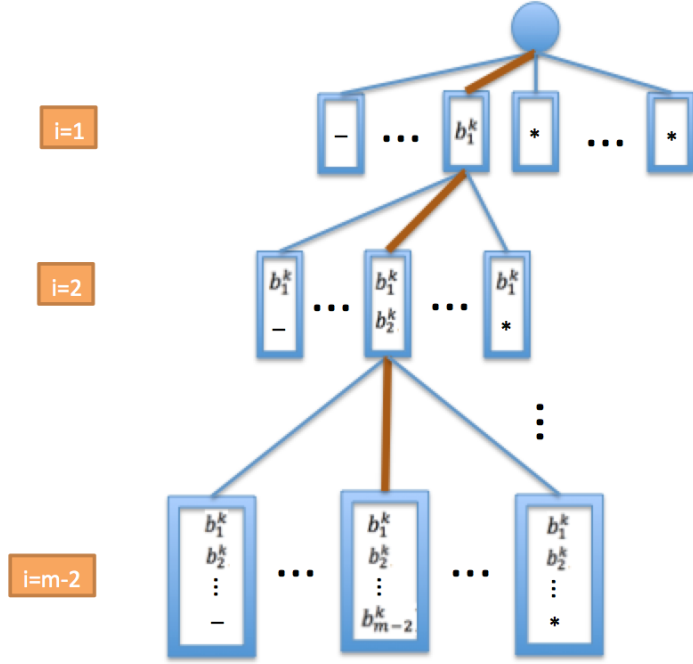
$$z_i \geq b_i^k, \quad i = 1, \dots, m$$

$$\mathbf{z} \in \mathbf{Z}$$

(P_S) probleminin optimal çözümü \mathbf{z}^* olsun. (P_{S_k}) probleminin optimal çözümü \mathbf{z}^k ise, $\mathbf{z}^* = \operatorname{argmax}_{k \in \mathbf{K}} wz^k$ olur. Bu noktada, (P_{S_k}) problemi olurlu bir çözüm içeriyorsa, bulunan çözüm, \mathbf{z}^k , baskın bir noktadır. Dolayısı ile $(n + 1)$. baskın çözümü bulmak için çözülen alt problemler sonucu ortaya birden fazla yeni baskın nokta çıkabilmektedir. Bu noktalardan $\mathbf{z}^k = \mathbf{z}^*$ olan çözüm haricindekiler optimal çözümü oldukları alt sınır vektörü, \mathbf{b}^k , ile birlikte saklanır. Bu alt sınır vektörü-baskın nokta ikilileri ve olurlu çözüm bulundurmayan alt sınır vektörleri sonraki döngülerde çözülmesi gereken alt problem sayısını azaltmamızı sağlayacaktır.

Yöntemimizde, $w_m = 1$ ve $w_i = \epsilon > 0, i = 1, \dots, m - 1$ değerlerine sabitlenmektedir. Her döngüde çözülen alt problemler sonucu bulunması muhtemel birden fazla baskın çözümden sadece wz değeri en yüksek olan çözüm \mathbf{R} kümesine eklendiği için, yeterince küçük ϵ değerleri ile j . baskın çözümün m . amaç fonksiyonu değeri, m . amaç fonksiyonu için j . en yüksek çözüm olmaktadır. Baskın noktaların üretilmesinde böyle bir sıranın takip edilmesi, arama uzayında kullanılan alt sınır vektörlerinin boyutlarını bir azaltmamızı sağlamaktadır, $\mathbf{b}^k \in \mathbb{R}^{m-1}$, $\mathbf{b}^k = (b_1^k, b_2^k, \dots, b_{m-1}^k)$.

$(n + 1)$. baskın noktanın m . amaç fonksiyonu değeri \mathbf{R} kümesindeki noktaların m . amaç fonksiyonu değerlerinden daha iyi olmayacağı garantilendiğinden, $(n+1)$. baskın nokta \mathbf{R} kümesindeki her noktadan diğer kriter değerlerinden en az birinde daha iyi olmak zorundadır. $(n + 1)$. baskın nokta \mathbf{z}^1 den 1. kriterde daha iyi ise, o zaman \mathbf{R} kümesindeki 1. kriter değeri \mathbf{z}^1 'den daha iyi olan noktalardan da $i = 2, \dots, m - 1$ kriter değerlerinden en az birinde daha iyi olmalıdır. $(n + 1)$. baskın nokta \mathbf{z}^2 'den 2. kriterde daha iyi ise, \mathbf{R} kümesindeki 1. kriter değeri \mathbf{z}^1 'den ve 2. kriter değeri \mathbf{z}^2 'den daha iyi olan noktalardan da $i = 3, \dots, m - 1$ kriter değerlerinden en az birinde daha iyi olmalıdır. Bu gerçekten yola çıkarak, mevcut \mathbf{R} kümesi var iken, $(n + 1)$. baskın noktayı bulmak için çözülmesi gereken alt problemlerde kullanılacak alt sınır vektörlerini aşağıdaki şekilde tanımlıyoruz:



Şekil 1. Alt sınır değerlerinin ağaç veri yapısı ile gösterimi

(P_{S_k}) problemini tanımlamak için \mathbf{R} kümesinin $(m-2)$ elemanlı bir alt kümesini kullanıyoruz. Varsayalım ki $\mathbf{k} = (k_1, \dots, k_{m-2})$ vektöründe k_i elemanı i . alt sınır değeri için \mathbf{z}^{k_i} baskın noktasının i . kriter değerinin kullanıldığını gösterebiliriz. Ek olarak, şu kümeleri tanımlayalım:

$$\mathbf{R}_n^{k_1} = \{ \mathbf{z}^j \in \mathbf{R} \mid z_1^j > z_1^{k_1} \}, \mathbf{R}_n^{k_i} = \{ \mathbf{z}^j \in \mathbf{R}_n^{k_{i-1}} \mid z_i^j > z_i^{k_i} \}, i = 2, 3, \dots, m-2.$$

O zaman, seçilecek \mathbf{k} vektörünün bulunduğu küme şu şekilde tanımlanabilir:

$$\{ \mathbf{k} \mid z^{k_1} \in \mathbf{R}, z^{k_i} \in \mathbf{R}^{k_{i-1}}, i = 2, \dots, m-2 \}.$$

Seçilen \mathbf{k} vektörünün tanımladığı (P_{S_k}) problemindeki alt sınır vektörü, \mathbf{b}^k ise şu şekilde elde edilir: $b_i^k = z_i^{k_i} + \epsilon, i = 1, \dots, m-2$ ve $b_{m-1}^k = \max_{\mathbf{z}^j \in \mathbf{R}, z_i^j \geq b_i^k, i=1, \dots, m-2} \{ z_m^j + \epsilon \}$.

(P_{S_k}) problemlerine karşılık gelen alt sınır vektörlerini verimli bir şekilde üretmek, saklamak ve sonraki döngülerde kolayca güncelleyebilmek için özel bir ağaç veri yapısını kullanıyoruz ("n-ary tree"). Bu ağaç yapısında $(m-2)$ seviye bulunmaktadır ve i . seviyede 1'den i 'ye kadar olan alt sınır değerleri tutan düğümler bulunmaktadır. i . seviyedeki kardeş düğümler (bir üst seviyede aynı ebeveyn düğüme bağlı düğümler), 1'den $i-1$ 'e kadar aynı alt sınır değerine sahip olup, b_i^k değerleri soldan sağa doğru artan şekilde sıralı biçimde tutulmaktadır. $(m-2)$ seviyesinde b_{m-1}^k belirlenerek, (P_{S_k}) problemi tanımlanmış olmaktadır. Şekil 1'de örnek bir ağaç gösterimi verilmektedir.

Şekil 1'deki gösterimde, (-) sembolü, ilgili yerde geçerli bir alt sınır bulunmadığını, (*) ise herhangi bir alt sınır değerini temsil etmek için kullanılmıştır. Her bir yaprak düğüm $(m-2)$ seviyesinde düğümler, (P_{S_k}) probleminin çözülmesi için gerekli alt sınır değerlerini içermektedir.

Baskın nokta içermediği bilinen veya hali hazırda bir baskın nokta içeren düğümler haricindeki tüm yaprak düğümler için (P_{S_k}) problemi çözülür.

Varsayalım ki, mevcut \mathbf{R} kümesi var iken bir sonraki baskın çözüm y olsun. Bu durumda, alt sınır ağacı şu şekilde güncellenir:

1. Alt sınır ağacının her seviyesinde:

(a) i . seviyedeki her kardeş düğüm kümesi için:

i. Eğer y çözümünün 1'den $(i - 1)$ 'e kadar olan kriter değerleri anne düğümdeki alt sınır değerlerinden (b_1, \dots, b_{i-1}) büyük veya eşit olup, $y_i + \epsilon$ değerine eşit i . alt sınır değerine sahip herhangi bir düğüm yok ise $y_i + \epsilon$ değerine sahip yeni bir düğüm yaratılır. Bu düğüm, soldan sağa artan sırayı bozmayacak şekilde kardeş düğüm kümesine eklenir. Bu düğüm, yaprak düğüm değil ise çocuk düğümü yaratılır ve alt sınır vektörü olarak $(b_1, \dots, b_{i-1}, y_i + \epsilon, -)$ değeri verilir.

2. Eğer y çözümü k yaprak düğümündeki 1'den $(m - 2)$ 'ye kadar olan alt sınırlardan büyük veya eşitse, o zaman k yaprak düğümündeki b_{m-1}^k alt sınır değeri güncellenir.

Geliştirilen ve detayları üstte tartışılan tüm baskın noktaları bulan bu algoritma (TBA) bu yaklaşım altta aşama aşama özetlenmektedir.

Algorithm 1 TBA

Aşama 1 Başlangıç. $\mathbf{R} = \emptyset$, $n = 1$, $w_m = 1$ ve $w_i = \epsilon > 0$ $i = 1, \dots, m - 1$ olarak ilk değerleri tanımlanır.

Aşama 2 Yeni baskın noktanın, z^n , bulunması.

Aşama 2.1. Ağaç veri yapısı güncellenerek (P_S) modeli, ağaç veri yapısında farklı alt sınır vektörleri ile tanımlanan (P_{S_k}) alt modellere bölünür.

Aşama 2.2. (P_{S_k}) problemleri çözülerek, olurlu problemlerin her biri için optimal z^k noktaları bulunur. Alt problemlerin hiçbirisi olurlu çözüm içermiyorsa, Aşama 3 ile algoritma durdurulur. En az bir çözüm olması durumunda $n = n + 1$ değeri güncellenerek Aşama 2.3 ile devam edilir.

Aşama 2.3. (P_S) probleminin optimal çözümü, z^* , olurlu alt model çözümleri kullanılarak bulunur: $z^n = z^* = \operatorname{argmax}_{k \in K} w z^k$.

Aşama 2.4. \mathbf{R} seti güncellenir, $\mathbf{R} = \mathbf{R} \cup z^n$. Aşama 2 tekrarlanır.

Aşama 3 Durma. Algoritma durdurulur. $\mathbf{R} = \{z^1, z^2, \dots, z^n\}$ kümesi tüm baskın nokta kümesini, $N = n$ değeri tüm baskın nokta sayısını gösterir.

Önerdiğimiz bu yöntem, C++ dilinde kodlanarak paket program haline getirilmiş olup, hazırlanan arayüz ile diğer programlar tarafından çağrılabilir. Alt sınır ağacında tanımlanan tek amaçlı problemlerin çözümü için IBM ILOG CPLEX Concert Technology kütüphanesi kullanılmıştır.

Yöntemimizin farklı problem parametrelerine bağlı olarak performansının nasıl değiştiğini gözlemlemek amacı ile, farklı amaç fonksiyonu sayılarında ve problem büyüklüklerindeki sırt çantası problemleri üzerinde testler yapılmıştır. Kullanılan test örnekleri kümesi Tablo 1'de verilmektedir. Her test örneği, geliştirilen web uygulaması aracılığı ile sunucu makinesindeki çözücüye verilmiş ve çözücü tarafından verilen problem için bulunan baskın nokta sayısı (N), toplam

Tablo 1. Test kümesi

| Örnek No | Amaç Fonksiyonu Sayısı (m) | Sırt Çantası Sayısı | İkili Değişken Sayısı |
|----------|--------------------------------|---------------------|-----------------------|
| 1 | 3 | 1 | 50 |
| 2 | 3 | 1 | 100 |
| 3 | 3 | 3 | 50 |
| 4 | 3 | 3 | 100 |
| 5 | 4 | 1 | 25 |
| 6 | 5 | 1 | 20 |

Tablo 2. Test sonuçları

| Örnek No | N | M | M/N | T (Saniye) |
|----------|------|------|-------|--------------|
| 1 | 405 | 884 | 2,18 | 29,94 |
| 2 | 2752 | 5175 | 1,88 | 348,06 |
| 3 | 784 | 1645 | 2,10 | 175,07 |
| 4 | 2790 | 5400 | 1,94 | 1559,98 |
| 5 | 202 | 1140 | 5,64 | 27,54 |
| 6 | 220 | 6311 | 28,69 | 1694,77 |

çözülen tek amaçlı problem sayısı (M) ve çözüm süresi (T) raporlanmıştır. Tablo 2'de sonuçlar sunulmaktadır.

Bu algoritma kullanılarak geliştirilen internet-tabanlı uygulama Bölüm 4.1'de, noktaların karar vericiye sunulması için geliştirilen görselleştirme araçları ise Bölüm 4.2'de sunulmaktadır.

3.2 Çok Amaçlı Tamsayı Problemlerinde (ÇATP) Dağılıma Bağlı Temsili Kümelerin Bulunması

Bölüm 3.1’de internet tabanlı olarak geliştirilen yaklaşımımız tüm baskın noktaların bulunmasını garantilemektedir ve tüm araştırmacıların kullanımına açık olarak tasarlanmıştır. Öyle ki, geliştirilen yeni sezgisel algoritmaların performansını ölçmek için de bir araç sunulmaktadır. Ancak, uygulamada yapılan denemeler bu aracın büyük ölçekli problemlerde çözüm zorluğu nedeni ile pratik olmadığını göstermiştir. İnternet tabanlı aracımızda bu nedenle çözüm süresi için bir üst sınır konmuştur. Ancak, bu süre içinde üretilecek kümenin kalitesi hakkında bir şey söylemek mümkün değildir. İnternet tabanlı uygulamada bu nedenle Ceyhan vd. (2019) tarafından geliştirilen temsili alt küme bulan yaklaşımlar da aracımız içine eklenmiştir.

Ancak; Sylva ve Crema (2007), Masin ve Bukchin (2008), ve Ceyhan vd. (2019) gibi mevcut yaklaşımlar, çözümlerin amaç fonksiyonu uzayında dağılım özelliklerini temsil edememektedir. Bu nedenle bu bölümde amaç, verilen bir ÇATP için eşit dağılılan bir temsili nokta kümesi yerine, kapsama hatası ile beraber yoğunluk özelliklerini dikkate alan temsili kümeler üretebilmektir. Özarık (2017)’in yüksek lisans tezi çalışmaları kapsamında yapılan bu çalışmada, ilk olarak ÇATP’ler için yoğunluk analizleri yapılarak dağılım özellikleri incelenmiştir.

3.2.1 Parzen Pencereleme Yaklaşımı ile Yoğunluk Tahmini

Parzen Pencereleme Yaklaşımı (Parzen Windows Approach - PWA), parametrik olmayan yoğunluk tahmini için sıkça kullanılan bir sınıflandırma tekniğidir (Duda, 1973, sayfa 164-177). Verilen bir çekirdek işlevi ile, gözlemlenen noktaların doğrusal bir birleşimi yoluyla belirli bir eğitim seti dağılımına (yoğunluğa) yaklaşır. Günümüzde örüntü tanıma, sınıflandırma ve görüntü bölütleme gibi uygulamaları vardır.

Bir z noktası etrafındaki Parzen yoğunluk değerini bulmak için aşağıdaki formül kullanılır:

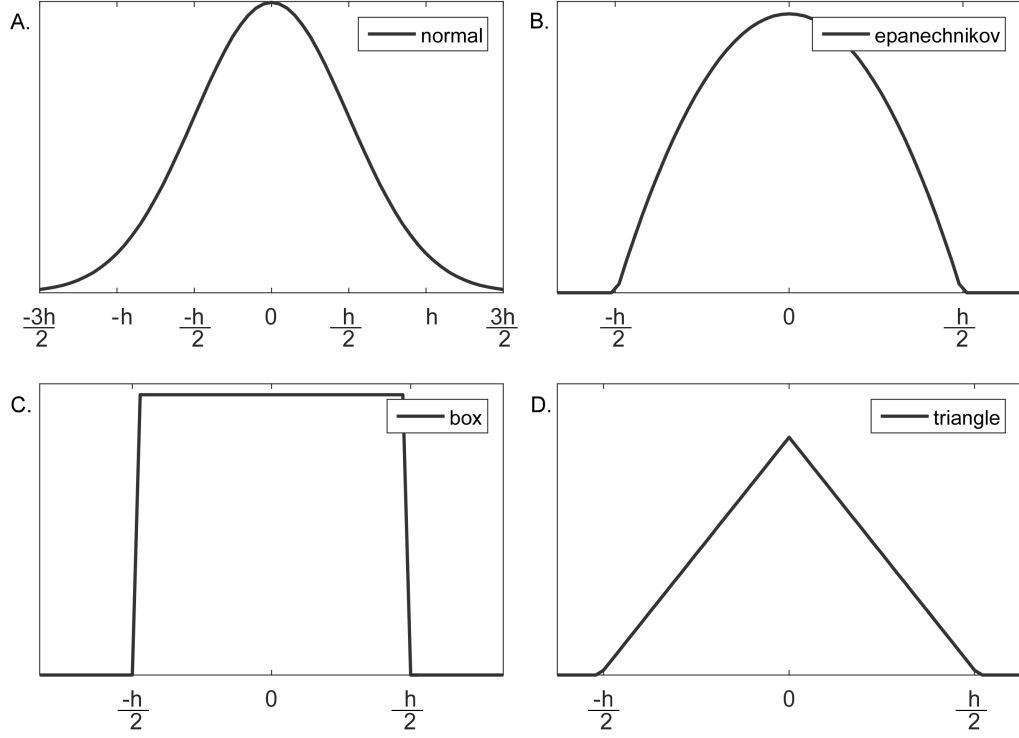
$$P(\mathbf{z}) = \frac{1}{hN} \sum_{n=1}^N K\left(\frac{d(\mathbf{z}^n, \mathbf{z})}{h}\right) \quad (4)$$

Burada h pencere açıklığını, $K(\cdot)$ ise \mathbb{R}^m uzayındaki pozitif bir pencere fonksiyonunu (kernel function) belirtmektedir ve alttaki koşulları sağlamaktadır:

$$\int_{-\infty}^{+\infty} K(u) du = 1 \quad (5)$$

$$K(u) = K(-u) \quad \forall u \quad (6)$$

Bizim çalışmamız kapsamında, $K(\cdot)$ fonksiyonu verilen bir noktanın etrafında yoğunluk hesabı yapılırken yoğunluğa katkısı olan komşu noktaların tanımlanması ve uzaklığına göre katkısının belirlenmesinde kullanılmaktadır. Yaygın olarak kullanılan fonksiyonlar Şekil 2’de sunulmuştur.



Şekil 2. Yaygın olarak kullanılan çekirdek işlevleri (kernel functions)

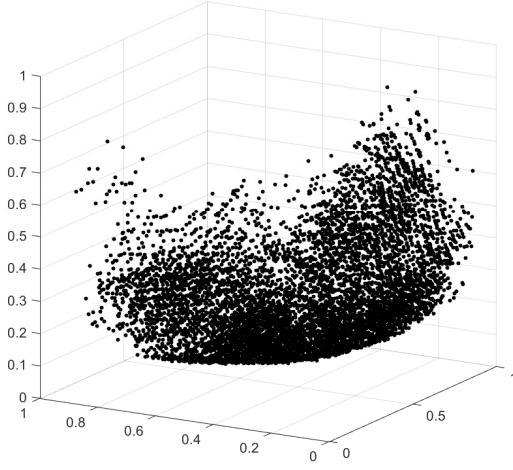
3.2.2 ÇATP'lerin Dağılım Özellikleri

ÇATP'ler için baskın nokta kümelerinin dağılım özellikleri incelenirken PWA yöntemi kullanılmıştır. Burada, daha önce tüm baskın nokta kümesi bilinen problemler üzerinde analizler yapılmıştır. Farklı ÇATP'lerdeki yoğunluk karşılaştırmalarının ve aynı problem içerisindeki farklı kriterlerin karşılaştırılmasının doğru bir şekilde yapılmasını sağlamak için, ilk olarak her bir baskın noktanın amaç değerlerinin aynı aralıkta olacak şekilde ölçeklendirilmesi gerekmektedir. Bu ölçeklendirme, Köksalan ve Lokman (2009) çalışmasındaki gibi yapılmıştır: $sz_i = \frac{z_i - z_i^{IP}}{z_i^{NP} - z_i^{IP}}$ $i = 1, \dots, m$.

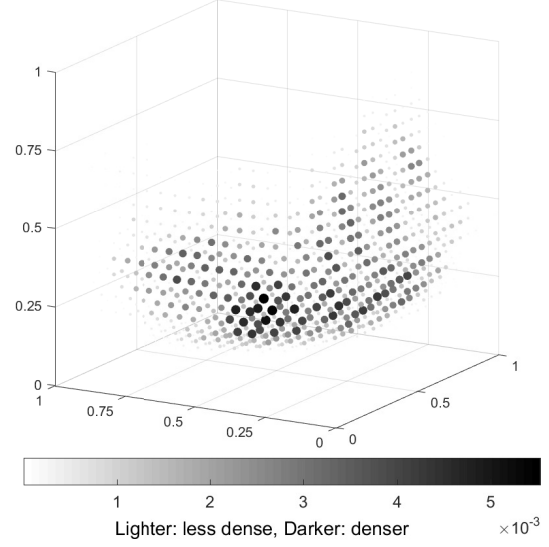
Daha sonra, ölçeklendirilmiş amaç fonksiyonu uzayı kenar uzunluğu e olan eşit büyüklükteki hiperküplere bölünerek, bu hiperküplerin merkez noktaları yoğunluk analizinde kullanılacak referans noktaları \mathbf{r}^j $j = 1, \dots, J$ olarak seçilmiştir ($J = \frac{1}{e^m}$). Yapılan analizlerde Şekil 2-D'de gösterilen üçgen çekirdek işlevi kullanılmıştır. Bu sayede, seçilen bir referans noktası etrafında yoğunluk analizi yapılırken sadece h komşuluğunda yer alan noktaların yoğunluğa etkisi vardır ve bu noktaların referans noktasına olan uzaklığı arttıkça yoğunluğa olan katkıları doğrusal olarak azalmaktadır.

$$K\left(\frac{d(\mathbf{sz}, \mathbf{r}^j)}{h}\right) = \begin{cases} 1 - \frac{d(\mathbf{sz}, \mathbf{r}^j)}{h/2}, & \text{eğer } \frac{d(\mathbf{sz}, \mathbf{r}^j)}{h} \leq 1/2 \\ 0, & \text{diğer} \end{cases} \quad (7)$$

Farklı büyüklükte farklı problemler tiplerinde karşılaştırılabilir bir yoğunluk ölçüsü tanımlamak amacı ile her referans nokta için bulunan yoğunluk değerleri önce toplanır daha sonra bu toplam



a. The Nondominated Points



b. PWA Density Values

Şekil 3. Bir ÇAAP problemi için baskın nokta kümesi ve PWA yoğunluk değerleri ($N = 6573$; $h = 0,2$; $e = 0,05$)

kullanılarak her biri, toplamı 1 olacak şekilde ölçeklendirilir:

$$P(\mathbf{r}^j) = \sum_{n=1}^N K \left(\frac{d(\mathbf{sz}^n, \mathbf{r}^j)}{h} \right) \quad (8)$$

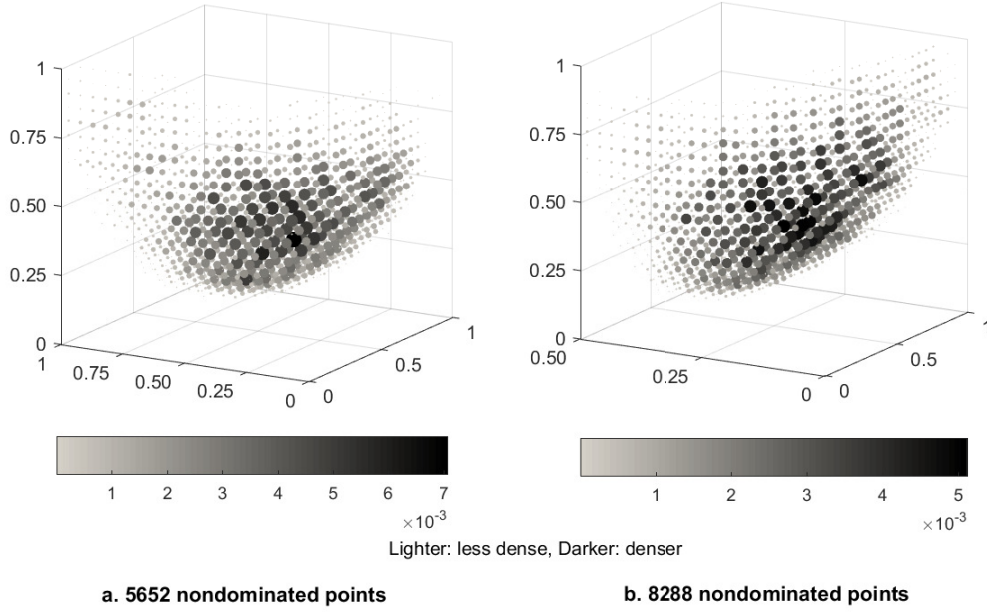
$$\bar{P}(\mathbf{r}^j) = \frac{P(\mathbf{r}^j)}{\sum_{j=1}^J P(\mathbf{r}^j)} \quad (9)$$

Burada, 4'te verilen denklemden farklı olarak toplam katkı, toplam nokta sayısı N ve pencere açıklığına, h , bölünmemektedir. Çünkü, tüm analiz hesaplamalarında yukarıdaki gibi normalizasyon yapılmış, karşılaştırılabilir hale getirilmiştir.

Şekil 3, Çok Amaçlı Atama Problemi (ÇAAP) için PWA Yöntemi ile yapılan yoğunluk analizi göstermektedir. Bu örnekte, referans noktaları üretilirken, amaç fonksiyonu uzayın kenar uzunluğu $e = 0,05$ olan hiperküplere bölünmüş ve pencere aralığı $h = 0,2$ olarak seçilmiştir. Yapılan analizlerde Öklid Uzaklık ölçüsü kullanılarak yoğunluk hesabı yapılmıştır. Şekil 3'te yoğunluk noktalarının büyüklüğü ve rengi ile gösterilmiştir. Öyle ki, noktalar büyüdükçe ve rengi griden siyaha yaklaştıkça yoğunluk artmaktadır.

Farklı ÇATP'lerde PWA yöntemi ile analizler yapılarak ortak özellikler araştırılmıştır. Şekil 4 farklı büyüklükteki Çok Amaçlı Sırt çantası Problemleri (ÇASP) için yoğunluk analizini gösterirken Şekil 5 benzer sayıda baskın noktası bulunan farklı ÇATP'ler için PWA ile elde edilen yoğunluk değerlerini göstermektedir. Yapılan tüm incelemeler, yoğunluğun merkeze gittikçe arttığını, uç noktalarda ise sifıra yaklaştığını göstermektedir.

PWA kullanılarak yapılan ön çalışmalarda, amaç fonksiyonu uzayı hiperküplere bölünmekte



Şekil 4. Sırasıyla 5652 ve 8288 baskın noktaya sahip 3-amaçlı 100-parçalı ÇASP'ler için PWA Yoğunluk Değerleri ($h = 0, 2$; $e = 0, 05$)

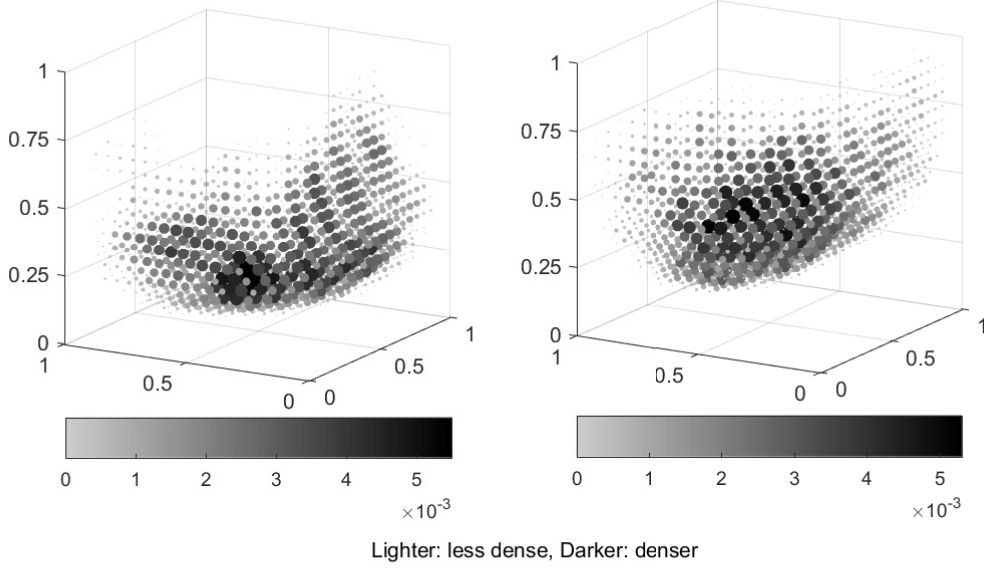
ve her bir hiperküpün merkez noktası referans noktası seçilerek yoğunluk hesabı yapılmaktadır. Ancak, yapılan incelemeler baskın nokta kümesinin bir yüzey oluşturduğunu ve bu bölgeler dışındaki bölgelerde baskın nokta bulunmadığını göstermektedir. Örneğin, Şekil 4 ve 5'te bir çok referans noktasında hesaplanan yoğunluk değeri sıfıra eşittir. Bu nedenle, tüm hiperküpler yerine baskın noktaların olası yerlerini belirleyerek; uzayın sadece bu bölümlerini incelemek daha etkin bir analiz yapmak ve işlem sayısını azaltabilmek açısından önemlidir. Bu sebeple, çalışmaların geri kalanında baskın nokta kümesini yaklaşık olarak tanımlayan Köksalan ve Lokman (2009) tarafından önerilen L_p yüzeyi üzerinde yoğunluk analizlerine devam edilecektir.

3.2.3 L_p Yüzeyi Üzerinde PWA Yoğunluk Analizi

Köksalan (1999), iki kriterli çizelgeleme problemleri için baskın nokta kümesini L_p yüzeyi kullanarak tanımlamıştır. Daha sonra, Köksalan ve Lokman (2009) bu çalışmayı genelleştirerek çok amaçlı bileşi problemleri için baskın nokta kümesini baskın nokta kümesinin tanımlanmasında kullanmışlardır. Bu çalışmalardaki deneyler, baskın nokta kümesinin yaklaşık olarak tanımlanmasında L_p yüzeyinin performansının iyi olduğunu göstermektedir. Bu nedenle, bu çalışmamızda, baskın nokta kümesinin yaklaşık olarak tanımlanmasında L_p yüzeyi kullanılmaktadır:

$$L_p = \left\{ \mathbf{sz} \in \mathbb{R}_{\geq}^m ; \left[\sum_{i=1}^m (1 - sz_i)^p \right]^{1/p} = 1 \quad p > 0 \right\} \quad (10)$$

Burada, Köksalan ve Lokman (2009) tarafından önerilen yöntemde olduğu gibi "merkezi" bir referans noktasından geçen L_p yüzeyine karşılık gelen p değeri kullanılmaktadır. Merkez olarak tanımlanan bu nokta, ideal noktaya Tchebycheff uzaklığı en küçük nokta olarak belirlenmiştir.



**a. 3-objective 30X30 MOAP with
6573 nondominated points**

**b. 3-objective 100-item MOKP with
6500 nondominated points**

Şekil 5. Benzer sayıda baskın noktası olan ÇAAP ve ÇASP için PWA yoğunluk değerleri ($h = 0, 2, e = 0, 05$)

L_p yüzeyi elde edildikten sonra yüzeyin farklı noktalarındaki yoğunluk değerini bulabilmek için, bu yüzey üzerine yaklaşık olarak eşit aralıklarla dağılan hipotetik noktalar yerleştirilmektedir. Bu noktaları yerleştirmek için bütün uzay önceki bölümlerde yapıldığı gibi her bir kenarı e birim olan küçük hiperküplere bölünür ve L_p yüzeyi üzerinde küplerin merkezi olan referans noktalara Öklid uzaklığı en yakın olan nokta bulunur. Bunu yapabilmek için, her bir referans noktası için \mathbf{r}^j noktası için L_p yüzeyi üzerinde Öklid uzaklığı en yakın olan \mathbf{q}^j noktası ($P_1(j)$) çözümlenerek bulunur.

($P_1(j)$):

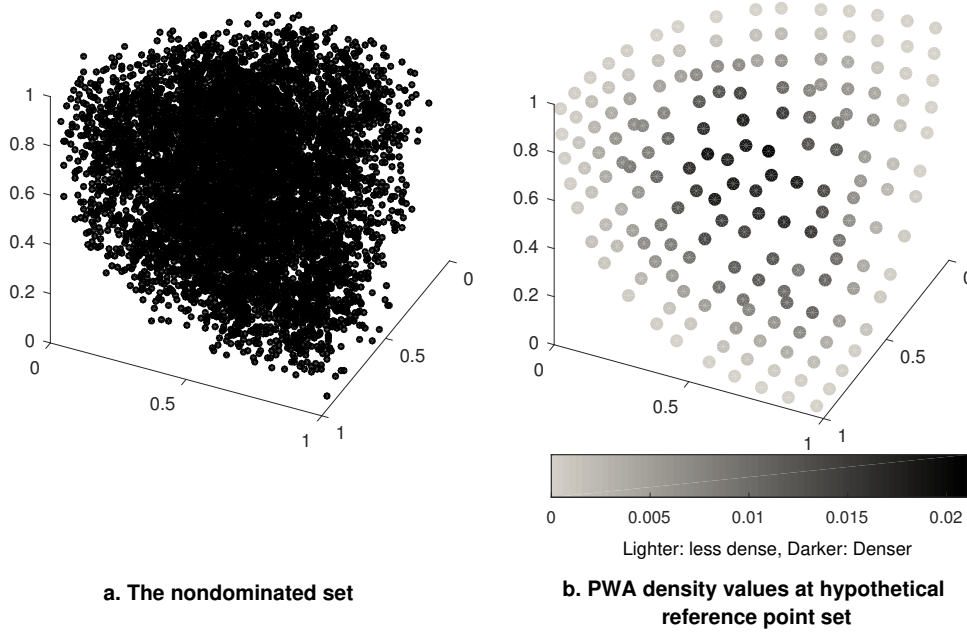
$$\min d(\mathbf{q}^j, \mathbf{r}^j) = \sqrt{\sum_{i=1}^m (r_i^j - q_i^j)^2} \quad (11)$$

$$\text{Kısıtlar} \quad \sum_{i=1}^m (1 - q_i^j)^p = 1 \quad (12)$$

$$0 \leq q_i^j \leq 1 \quad i = 1, \dots, m \quad (13)$$

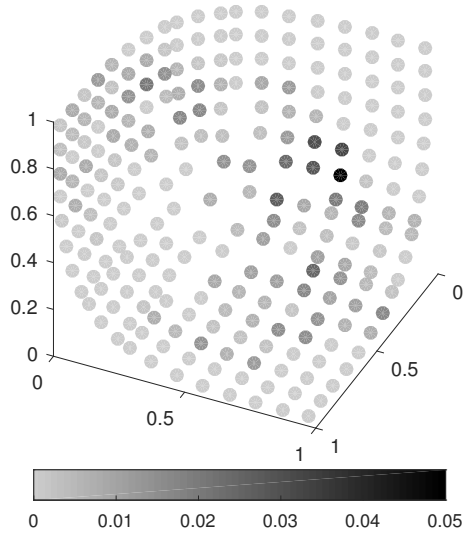
$d(\mathbf{q}^j, \mathbf{r}^j) > \frac{\epsilon}{2}$ durumunda, referans noktanın, \mathbf{r}^j , L_p yüzeyinden uzak olması nedeni ile yoğunluk analizinden çıkarılır. Aksi takdirde, ona yakın olan L_p yüzeyindeki \mathbf{q}^j noktası, hipotetik referans nokta küme setine eklenir: $\mathbf{Q} = \{\mathbf{q}^j : d(\mathbf{q}^j, \mathbf{r}^j) \leq \frac{\epsilon}{2} \quad j = 1, \dots, J\}$. Hipotetik küme seti elde edildikten sonra, PWA ile yoğunluk analizi sadece bu noktalar için yapılmaktadır.

Şekil 6, 3-amaçlı ÇASP için hipotetik referans noktaları üzerinde PWA ile hesaplanan yoğunluk değerlerini göstermektedir. L_p yüzeyi üzerinde çalışmak, baskın nokta olmayan bölgelerdeki gereksiz işlem maliyetlerini azaltmaktadır. Köksalan ve Lokman (2009) tarafından da tartışıldığı gibi, L_p yüzeyinin baskın noktaları temsili uç noktalara gitikçe kötüleşebilmektedir, ancak yoğunluğun da bu bölgelerde azalması nedeni ile PWA ile beraber kullanılması açısından önemli bir yöntemdir.

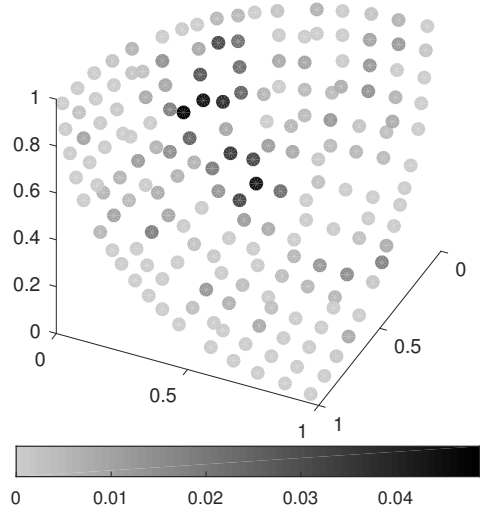


Şekil 6. Baskın nokta kümesi ve hipotetik referans nokta kümesi üzerinde PWA ile hesaplanan yoğunluk değerleri ($N = 6500$; $h = 0, 2$; $e = 0, 1$)

PWA yöntemi kullanılarak, yoğunluk değerlerinin farklı büyüklüklerdeki ve farklı tiplerdeki problemlerdeki değişimleri araştırılmıştır. Çok Amaçlı Sırt Çantası Problemleri (ÇASP), Çok Amaçlı En Kısa Patika Problemleri (ÇAKPP), Çok Amaçlı Minimum Kapsayan Ağaç Problemleri (ÇAMKAP) ve Çok Amaçlı Atama Problemlerini (ÇAAP) içeren 100 farklı boyuttaki ve tipteki problem ile çalışılmıştır. Çalışılan problemler, yaklaşık olarak 150 ile 11.000 baskın nokta içermektedir. Bu sayının 1000'den az olduğu örneklerde, Şekil 7'de de gösterildiği gibi baskın nokta kümesinin dağılım özellikleri birbirlerinden daha çok farklılık göstermektedir ve ortak bir yapı gözlenmemiştir. Baskın nokta sayısı arttıkça, örneğin baskın nokta sayısı 1000'in üzerinde olan problemlerde, farklı ÇATP'lerin aynı özellikler göstermeye başladığı gözlenmiştir. Şekil 8, 120 baskın noktaya sahip bir ÇAAP ile 6500 noktaya sahip bir ÇASP için yoğunluk değerlerini karşılaştırmaktadır. Bu şekillerde, yoğunluk arttıkça renk griden siyaha doğru koyulaşmaktadır. Yapılan bu analizlere göre, en yoğun bölge yüzeyin merkezinde yer alırken uçlara doğru ilerledikçe hipotetik noktaların yoğunlukları azalmaktadır. Şekil 9'de gösterildiği gibi bu özellikler sadece farklı büyüklükteki ÇATP'ler için değil aynı zamanda farklı tipteki problemler için de ortaktır.



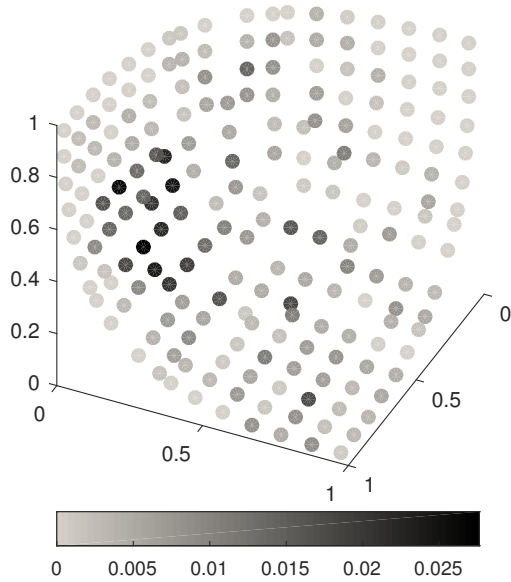
a. 103 nondominated points



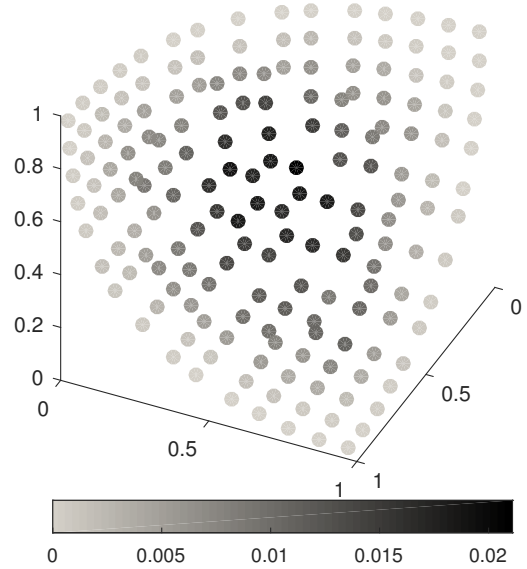
b. 107 nondominated points

Lighter: less dense, Darker: denser

Şekil 7. Küçük ölçekli problemlerde hipotetik referans nokta kümesi üzerinde PWA ile hesaplanan yoğunluk değerleri ($h = 0, 2; e = 0, 1$)



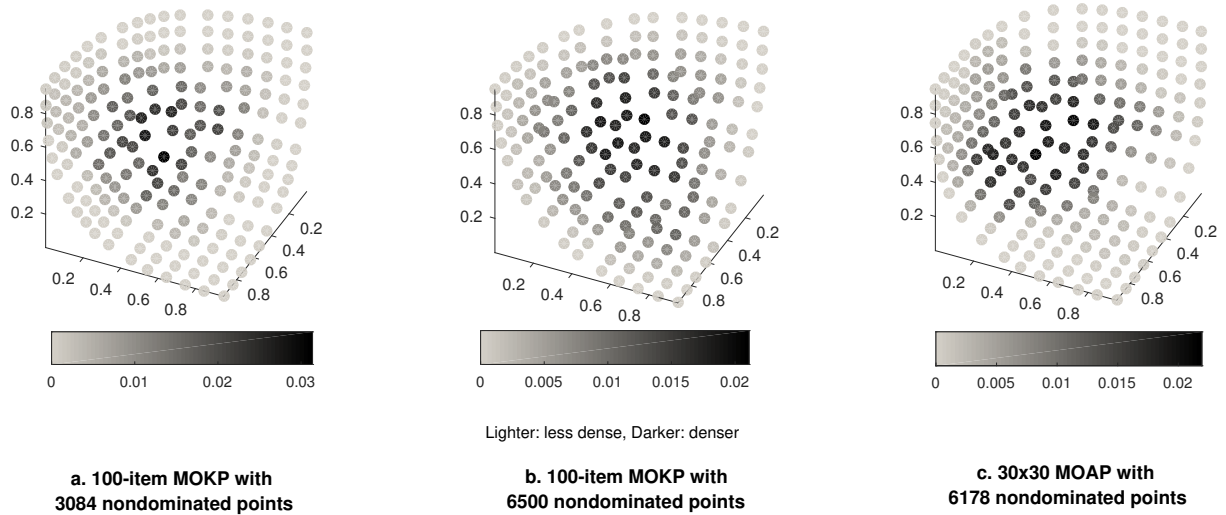
**a. 3-objective 10X10 MOAP with
120 nondominated points**



**b. 3-objective 100-item MOKP with
6500 nondominated points**

Lighter: less dense, Darker: denser

Şekil 8. Farklı büyüklükteki problemler için, hipotetik referans nokta kümesi üzerinde PWA ile hesaplanan yoğunluk değerleri ($h = 0, 2; e = 0, 1$)



Şekil 9. 3-amaçlı problemler için hipotetik referans nokta kümesi üzerinde PWA ile hesaplanan yoğunluk değerleri ($h = 0, 2$; $e = 0, 1$)

3.2.4 Hipotetik Noktaların Yoğunluk Değerlerine Göre Kategorilere Ayrılması

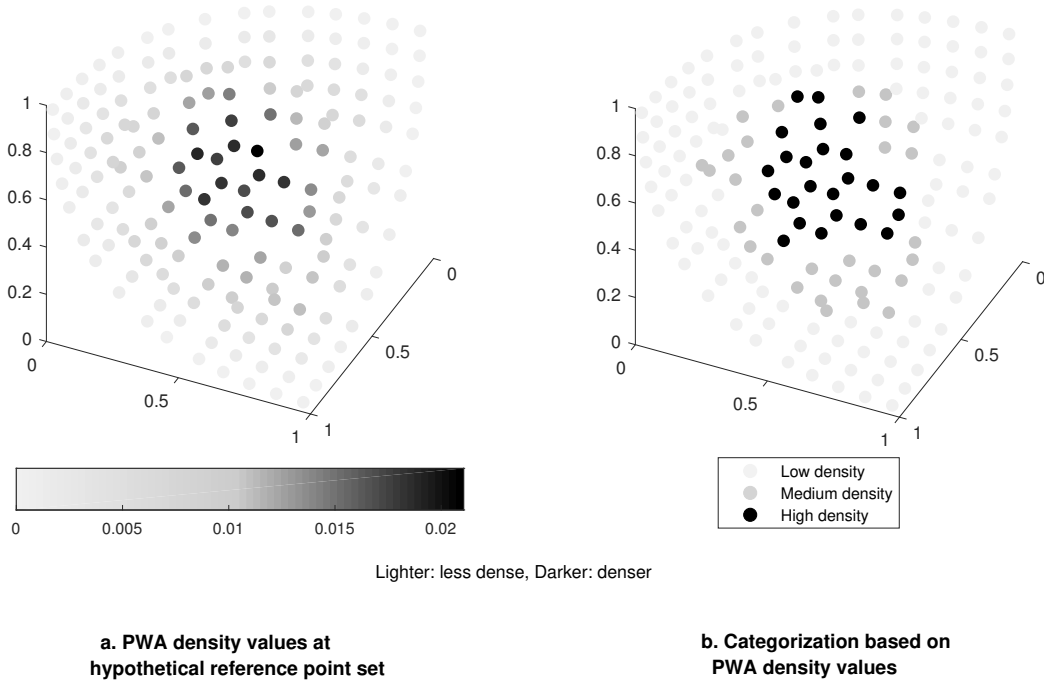
Bölüm 3.2.3'teki gözlemlerimiz doğrultusunda, bu bölümde ürettiğimiz hipotetik noktaları tahmin edilen yoğunluk değerlerine göre sınıflandıran yaklaşımlar geliştirilmiştir. Belirli bir dağılım özelliğine sahip olmadığı gözlemlenen baskın nokta sayısı 1000'den az olan problemler çalışma kapsamına alınmamıştır. Çözümü zor olan ve temsili küme bulmanın daha önem taşıdığı büyük ölçekli problemler üzerine yaklaşımlar geliştirilmiştir.

Baskın noktaları, C farklı kategoriye KT_1, \dots, KT_C atamak için üretilen hipotetik nokta kümesi kullanılmıştır. Baskın noktaların büyük bir kısmının hipotetik noktaların sadece küçük bir kısmı etrafında yer alması nedeni ile Pareto kuralına benzer bir sınıflandırma yapılmıştır. Burada, ilk olarak hipotetik noktalar yoğunluk değerlerine göre büyükten küçüğe sıralanmıştır. Yapılan incelemelerde, iki kategori için sınıflandırma yapıldığında, $C = 2$, ilk sıradaki yoğun bölgede yer alan noktaların %25'i toplam yoğunluk miktarının %70'inden sorumlu olduğu gözlenmiştir. Tablo 3'te de gösterildiği gibi toplam nokta sayısının %75'ini oluşturan geriye kalan noktalar ise sadece toplam yoğunluğun %30'unu oluşturmaktadır.

Tablo 3. Yoğunluk bazlı sınıflandırılan hipotetik noktaların kategorilere göre dağılımı, $C = 2$

| Problem Tipi | Kategori 1 | Kategori 2 |
|-----------------------|---------------------------|---------------------------|
| | (toplam yoğunluğun %70'i) | (toplam yoğunluğun %30'u) |
| ÇAAP | %26 | %74 |
| ÇAKPP | %25 | %75 |
| ÇASP | %24 | %76 |
| Genel Ortalama | %25 | %75 |

Aynı şekilde, hipotetik noktaları üç kategoriye ayırmak istediğimizde, $C = 3$, Tablo 4'de gös-



Şekil 10. 3-amaçlı 100-parçalı bir ÇASP için hipotetik noktaların PWA ile bulunan yoğunluk değerleri ve sınıflandırılması ($N = 6500$; $C = 3$; $h = 0, 2$; $e = 0, 1$)

terildiği gibi en yoğun noktaların sadece %11'i toplam yoğunluğun %40'ını oluşturmaktadır. Bu sınıflandırmada, geriye kalan noktalar içinde yer alan ve hipotetik noktaların %14'ünü oluşturan noktaların yoğunluğa katkısı %30, diğer noktaların katkısı ise %40 olarak hesaplanmıştır. Bu sınıflandırmada, Kategori 1 çok yoğun, Kategori 2 yoğun ve Kategori 3 az yoğun olarak tanımlanmıştır. Şekil 10, ÇASP için üretilen hipotetik noktaların sınıflandırmasını göstermektedir.

Tablo 4. Yoğunluk bazlı sınıflandırılan hipotetik noktaların kategorilere göre dağılımı, $C = 3$

| Problem Tipi | Kategori 1 (toplam yoğunluğun %40'ı) | Kategori 2 (toplam yoğunluğun %30'u) | Kategori 3 (toplam yoğunluğun %30'u) |
|-----------------------|---|---|---|
| ÇAAP | %11 | %15 | %74 |
| ÇAKPP | %11 | %14 | %75 |
| ÇASP | %10 | %14 | %76 |
| Genel Ortalama | %11 | %14 | %75 |

3.2.5 Yoğunluk Bazlı Tahmin Yöntemi (Density-based Approximation Algorithm - DAA)

Önceki bölümlerde elde edilen bulgular doğrultusunda, baskın nokta kümesi bilinmeyen herhangi bir ÇATP için baskın noktaların olası yerlerini ve yoğunluk sınıfını tahmin eden bir yöntem geliştirilmiştir. Bu yöntemde, baskın nokta kümesini yaklaşık olarak temsil eden ve yoğunluk tahminine göre kategorilere ayrılmış noktalar üretmek amaçlanmaktadır. Bu kapsamda, ilk olarak "merkezi" bir baskın noktaya göre L_p yüzeyi tanımlanmakta, daha sonra bu yüzey üzerinde hipotetik nok-

olar üretilmektedir. Baskın nokta kümesi bilinmediğinden, bu hipotetik noktalar gerçek yoğunlukları hesaplanamamaktadır. Merkezi olan tanımlanan noktanın en fazla yoğunluğa sahip nokta olduğu ve yoğunluğun bu noktadan uzaklaştıkça azalacağı varsayımı altında, hipotetik noktalar alttaki gibi sınıflandırılmaktadır:

- Merkez noktasına en yakın, hipotetik noktaların %11'i oluşturan noktalar Kategori 1'de yer alır. Kategori 1'in, toplam yoğunluğun %30'unu oluşturduğunu tahmin edilir.
- Geriye kalan noktalar içinde merkez noktasına en yakın olan, hipotetik noktaların %14'ünü oluşturan noktalar Kategori 2'ye alınır. Kategori 2, toplam yoğunluğun tahmini % 30'unu oluşturur.
- Toplam noktaların %75'ini oluşturan geriye kalan tüm noktalar, Kategori 3'te yer alır. Kategori 3'te yer alan noktalar, tahmini olarak yoğunluğun % 30'unu oluşturur.

Algorithm 2 Yoğunluk Bazlı Tahmin Yöntemi (DAA)

Aşama 1. h ve e değerleri belirlenir. $Q = \emptyset$ olarak tanımlanır.

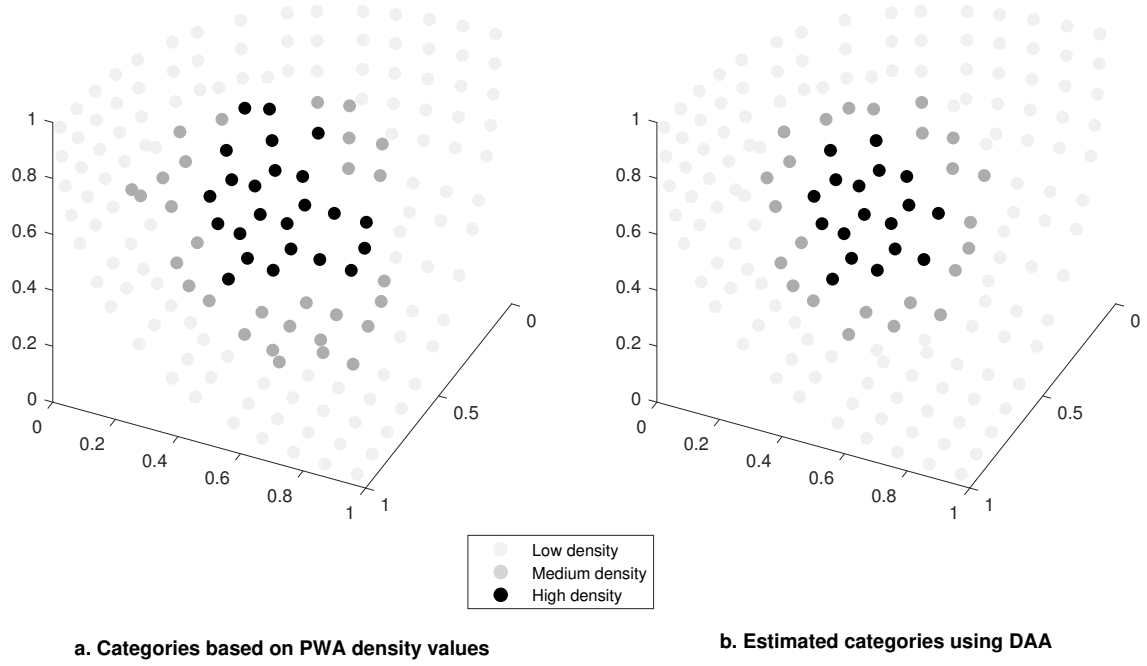
Aşama 2 L_p yüzeyinin Tanımlanması. İdeal noktasına en yakın n^* çözümü bulunur: $n^* = \arg \min_{n=1, \dots, N} \max_{i=1, \dots, m} sz_i^n$. Bu çözümden geçen L_p yüzeyinin bulunması için p değeri

hesaplanır: $\sum_{i=1}^m (1 - sz_i^{n^*})^p = 1$.

Aşama 3 Hipotetik Noktaların Üretilmesi. Amaç fonksiyonu uzayı kenar uzunluğu e birim olan hiperküllere bölünür. her bir hiperküp için merkez noktası r^j olarak tanımlanır ve $P_1(j)$ modeli çözülerek L_p yüzeyi üzerinde o noktaya en yakın olan q^j noktası bulunur. Eğer $d(q^j, r^j) < \frac{e}{2}$ koşulu sağlanıyorsa $Q = Q \cup \{q^j\}$ olarak güncellenir.

Aşama 4 Hipotetik Noktaların sınıflandırılması. Her bir hipotetik noktanın, $q^j \in Q$, merkez noktasına, sz^{n^*} , Tchebycheff uzaklığı hesaplanır. En yakında yer alan ve toplam hipotetik noktaların %11'ini oluşturan noktalar Kategori 1, onu takip eden ve noktaların %14'ünü oluşturan noktalar Kategori 2, ve geriye kalan tüm noktalar Kategori 3 olarak sınıflandırılır.

DAA'in performansını değerlendirmek için her bir hipotetik noktanın tahmin edilen yoğunluk kategorisi ile gerçek kategorileri karşılaştırılmıştır. Şekil 11, bir ÇASP için tahmini ve gerçek kategorileri karşılaştırmaktadır. 51 farklı ÇATP üzerinde yapılan deneylerde, DAA hipotetik noktaların ortalama %87'inde kategorileri doğru tahmin etmiştir. Diğer hipotetik noktalarda yapılan tahmin hatası bir yoğunluk seviyesinden fazla değildir. Hipotetik noktaların %6'sı gerçek kategorisinden bir seviye daha az yoğun kategoriye atanırken, %7'si bir seviye daha yoğun olarak sınıflandırılan bir kategoriye atanmıştır. Tüm bu sonuçlara göre DAA, hem az maliyetle, (merkezi bir baskın nokta üreterek ve yüzey tanımlayarak), hem de az hata ile baskın noktaları temsil eden ve yoğunluğuna göre sınıflandırılmış hipotetik noktalar üretmektedir.



Şekil 11. Yoğunluk bazlı sınıflandırılan hipotetik noktaların gerçek ve tahmin edilen kategorileri ($N = 6500$; $C = 3$; $h = 0, 2$; $e = 0, 1$)

3.2.6 Çok Amaçlı Tamsayı Problemleri (ÇATP) için Yoğunluk Bazlı Temsilci Kümelerinin Bulunması

DAA, verilen bir ÇATP için temsili hipotetik noktalar üretmekte ve bu noktaları kategorilerine göre sınıflandırmaktadır. Ancak; hipotetik noktalar yerine, karar vericiye gerçek noktalar yani uygulanabilir çözümler sunmak önemlidir. Bu aşamada, dağılım özelliklerini dikkate alan temsilci kümeleri üretmek (örneğin, yoğun olan bölgelerden daha çok, az yoğun bölgelerden daha az temsilci baskın nokta bulmak gibi) istenebilir. Burada, hem dağılım özelliklerini hem de karar vericinin tercihlerini yansıtan temsilci küme üretilirken kullanılacak yeni bir performans ölçüsüne ihtiyaç duyulmaktadır.

Bu kapsamda, ilk olarak herhangi bir temsili kümenin tüm baskın nokta kümesini temsil gücünü değerlendirirken yoğunluğa göre ölçümler yapan yeni bir kalite ölçüsü önerilmektedir. Bu kalite ölçüsü; Masin ve Bukchin (2008) tarafından kullanılan kapsama ölçüsüne göre hataları değerlendirirken, bölgelerin yoğunluk seviyelerine göre farklı ağırlıklar dahil edilerek tasarlanmıştır.

Tanım 10 (Ağırlıklı Kapsama Ölçüsü): $\mathbf{R} \subseteq \mathbf{Z}^B$ kümesi, \mathbf{Z}^B için temsili küme ve $0 \leq \lambda(\mathbf{z}) \leq 1$ baskın nokta \mathbf{z} için ağırlık değeridir. Her bir baskın nokta $\mathbf{z} \in \mathbf{Z}^B$ için, $\mathbf{R} \subseteq \mathbf{Z}^B$ temsilci kümesi içinde, $\lambda(\mathbf{z})d(\mathbf{y}, \mathbf{z}) \leq \alpha(\lambda)$ koşullarını sağlayan bir temsilci $\mathbf{y} \in \mathbf{R}$ var ise, \mathbf{R} kümesine \mathbf{Z}^B

kümesinin $d_{\alpha(\lambda)} - temsili$ denir. \mathbf{R} kümesinin ağırlıklı kapsama hatası ise alttaki gibi hesaplanır:

$$\alpha(\lambda) = \max_{z \in \mathbf{Z}^B} \min_{y \in \mathbf{R}} \lambda(z) d(y, z) \quad (14)$$

Kapsama hatası hesaplamalarında da olduğu gibi, her bir baskın nokta $z \in \mathbf{Z}^B$ için ona en yakın temsilci çözüm, $y \in \mathbf{R}$, atanmaktadır. Ancak, ağırlıklı kapsama ölçüsü hesaplanırken, her çözümde yapılan hatayı farklı ağırlıklar kullanarak hesaplamakta, ve daha sonra maksimum ağırlıklı kapsama hatası bulunmaktadır. Bu nedenle, $z \in \mathbf{Z}^B$ çözümünün ne kadar iyi kapsandığı onun için atanan $\lambda(z)$ ağırlığına bağlıdır.

Bu çalışmada ölçeklendirilmiş amaç fonksiyonu üzerinde S' çalışmakta olup uzaklık ölçüsü olarak Tchebycheff temelli bir ölçü kullanılmaktadır:

$$\alpha(\lambda) = \max_{sz \in S'} \left(\min_{sy \in R'} \left(\max_{1 \leq i \leq m} (\lambda(sz)(sy_i - sz_i)) \right) \right) \quad (15)$$

Burada kullanılan ölçeklendirme ile problemler minimizasyon problemine dönüştürülmektedir ve bu nedenle kapsama hatası hesaplanırken baskın nokta sz 'nin temsilcisi sy 'den daha iyi olduğu (yani daha küçük olduğu) kısım dikkate alınmaktadır.

Bu çalışmada ağırlık vektörü, KT_1, \dots, KT_C kategorilerine göre atanmaktadır. Öyle ki, sz noktası için seçilen ağırlık $\lambda(sz)$ etrafındaki çözüm yoğunluğuna bağlıdır:

$$\lambda(sz) = \begin{cases} \lambda_1 & \text{if } sz \in KT_1 \\ \lambda_2 & \text{if } sz \in KT_2 \\ \dots & \\ \lambda_C & \text{if } sz \in KT_C \end{cases} \quad (16)$$

Burada, $\lambda_1, \dots, \lambda_C$ kategorilerin özelliklerine ve karar vericinin tercihlerine göre seçilebilir. Örneğin, çok yoğun bölgelerden daha çok, az yoğun bölgelerden daha az çözüm bulmak isteniyorsa; çok yoğun bölgelerde az yoğun bölgelere kıyasla daha büyük ağırlıklar kullanılabilir. Bu sayede, çok yoğun bölgelerde yapılan kapsama hataları daha çok cezalandırılacak bu nedenle bu bölgelerden daha çok temsilci bulunması sağlanacaktır.

3.2.7 Dağılım-bazlı Temsilci Bulma Algoritması (A Distribution-based Representation Algorithm - DBA)

Verilen herhangi bir ÇATP için; dağılım özelliklerini karar vericini belirlediği kalite ölçüsüne göre yansıtan baskın nokta kümesi üretebilmek için, Dağılım-bazlı Temsilci Bulma Algoritması (DBA) geliştirilmiştir. Bu algoritmada, ilk olarak DAA kullanılmaktadır. Bu sayede, baskın nokta kümesini temsil eden ve yoğunluk özelliklerine göre kategorilere ayrılmış Q kümesi üretilmektedir. Daha sonra, (P_2) modeli çözülerek Q kümesinin $d_{\alpha(\lambda)} - temsili$ olan bir alt kümesi bulunmaktadır.

(P₂):

$$\min \sum_{j=1}^{|\mathbf{Q}|} v_j \quad (17)$$

$$\text{Kısıtlar } (q_i^j - q_i^l)\lambda(\mathbf{q}^l)u_{lj} \leq \alpha(\boldsymbol{\lambda}) \quad \forall i, j, l \quad (18)$$

$$\sum_{l=1}^{|\mathbf{Q}|} u_{lj} \leq |\mathbf{Q}|v_j \quad \forall j \quad (19)$$

$$\sum_{j=1}^{|\mathbf{Q}|} u_{lj} = 1 \quad \forall l \quad (20)$$

$$v_j, u_{lj} \in \{0, 1\} \quad i = 1, \dots, m \quad j, l = 1, \dots, |\mathbf{Q}| \quad (21)$$

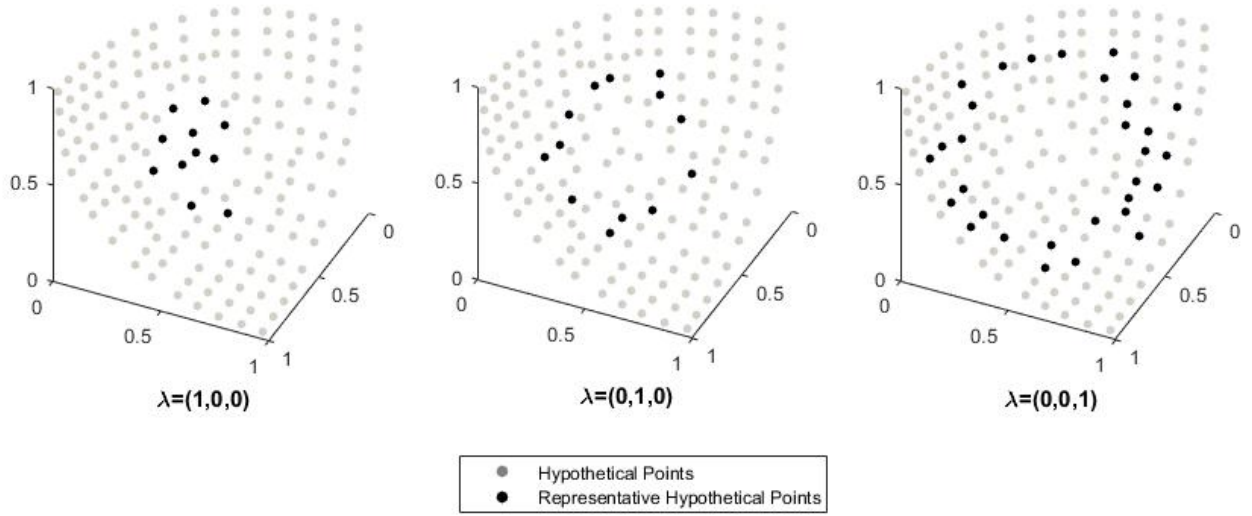
Bu modelde, ikili deęişken v_j eęer hipotetik nokta \mathbf{q}^j temsilci olarak seęilmiřse 1, aksi takdirde sıfır deęerini almaktadır. u_{lj} ikili deęişkeni ise, \mathbf{q}^l hipotetik noktası \mathbf{q}^j tarafından temsil ediliyorsa ancak 1 deęerini almaktadır. İlk kısıt, u_{lj} ikili deęişkeni 1 deęerini aldıęında etkin hale gelmektedir ve yapılan kapsama hatası Denklem 15 kullanılarak hesaplanmaktadır. Burada, temsilci olarak seęilen alt küme içinde yer alan hipotetik noktaların kümesi \mathbf{Q}^* , $\mathbf{Q}^* = \{\mathbf{q}^j \in \mathbf{Q} | v_j = 1\}$ olarak tanımlanmaktadır.

řekil 12, bir ÇASP için üretilen hipotetik noktaların sırasıyla $\boldsymbol{\lambda} = (1, 0, 0)$, $\boldsymbol{\lambda} = (0, 1, 0)$, ve $\boldsymbol{\lambda} = (0, 0, 1)$ aęırlık vektörlerine göre bulunan $d_{\alpha(\boldsymbol{\lambda})} - \text{temsilleri}'ni$ göstermektedir. Gösterildięi gibi, aęırlık vektörü $\boldsymbol{\lambda} = (1, 0, 0)$ olarak seęildięinde çok yoęun bölgelerde, $\boldsymbol{\lambda} = (0, 0, 1)$ olarak seęildięinde ise az yoęun bölgelerde daha çok temsilci bulunmaktadır.

Hipotetik temsili altküme, $\mathbf{Q}^* \subseteq \mathbf{Q}$, bulunduktan sonra, DBA her bir temsilci hipotetik nokta için ona Tchebycheff uzaklıęı en yakın geręek baskın noktayı, s_y , bulmaktadır:

$$s_y = \min_{\mathbf{sz} \in \mathbf{S}'} \max_{i=1, \dots, m} |sz_i - q_i| \quad (22)$$

DBA'nın performansını deęerlendirmek için, farklı ÇATP'lerde deneyler yapılmıřtır. Bulgular, Bölüm 4.3'te sunulmaktadır.



Şekil 12. 3198 baskın noktaya sahip 3-amaçlı, 100-parçalı bir ÇASP için hipotetik noktalar arasından temsilci seçimi

Algorithm 3 DBA

Aşama 1 Başlangıç. λ ve $\alpha(\lambda)$ karar verici tarafından belirlenir. $\mathbf{R} = \emptyset$ olarak tanımlanır.

Aşama 2 Hipotetik Nokta Üretilmesi ve Sınıflandırılması. DAA uygulanarak hipotetik nokta kümesi Q ve yoğunluk kategorileri bulunur.

Aşama 3 Temsilci Hipotetik Nokta Seçilmesi. (P_2) modeli çözülerek temsilci Q^* kümesi bulunur.

Aşama 4 Temsilci Baskın Nota Kümesinin Bulunması. Her bir $q \in Q^*$ için, ona en yakın baskın nokta bulunur: $sy = \min_{sz \in S'} \max_{i=1, \dots, m} |sz_i - q_i|$. Bulunan nokta, temsilci kümeye eklenir: $\mathbf{R} = \mathbf{R} \cup \{sy\}$.

Aşama 5 Durma. Algoritma durdurulur. \mathbf{R} temsili baskın nokta alt kümesini gösterir.

3.3 Çok Amaçlı Tamsayı Problemlerinde (ÇATP) Kullanıcı Etkileşimli Çözüm Yöntemlerinin Geliştirilmesi

Bölüm 3.2’de, ÇATP’ler için amaç fonksiyonu uzayında yoğunluk analizi yapılmış ve ortak dağılım özellikleri keşfedilmiştir. Buna göre karar verici tarafından verilen kapsama ölçüsünü yaklaşık olarak sağlayan ve dağılım özelliklerini dikkate alan baskın nokta kümesi bulan bir yöntem geliştirilmiştir. Pratik olan ve büyük problemlerde de uygulanabilecek bu yöntem, DBA, karar vericinin belirlediği ağırlıklı kapsama ölçüsüne yakın bir kalite seviyesi elde etmekle birlikte istenilen kapsama ölçüsünü garantilememektedir. Bu nedenle, bu bölümde karar verici tarafından verilen kapsama kalite ölçüsünü garantileyen temsili baskın çözümlerin bulunması amacıyla TSGA (Territory-Excluded Supported Generating Algorithm) ve TSGA-II olarak iki algoritma geliştirilmiştir. Doğan (2018)’in yüksek lisans tezi kapsamında geliştirilen bu algoritmalar da DBA gibi verilen herhangi bir ÇATP için dağılım özellikleri dikkate alarak altkümeler bulacak şekilde tasarlanmıştır. Farklı olarak, probleme özgü dağılım özelliklerini yansıtabilmek amacıyla her bir amaç fonksiyonu için algoritma boyunca kullanılacak ağırlıklar bulunmaktadır. Bu algoritmalar, Δ olarak tanımlanan kapsama ölçüsünü küçük alt kümeler üreterek garantilemektedir. Bu kapsamda geliştirilen yaklaşımlar Bölüm 3.3.1 ve 3.3.2’de sunulmaktadır. Yapılan deneyler ve bulgularımız ise Bölüm 4.4’te yer almaktadır.

3.3.1 TSGA (Territory-Excluded Supported Generating Algorithm)

Verilen bir kapsama ölçüsünü sağlayan temsili çözümlerin üretilmesi amacıyla literatürde bazı yaklaşımlar bulunmaktadır. Sylva ve Crema (2007) and Masin ve Bukchin (2008) tüm baskın çözümleri temsil edecek temsili bir altküme üretmek için algoritmalar önermişlerdir. Bu algoritmalarda ortak olan temel amaç, her adımda mevcut üretilmiş noktalar tarafından en kötü temsil edilen baskın noktayı üreterek temsili altküme eklemektir. En kötü temsil edilen baskın nokta, mevcut üretilmiş temsili çözümlere en uzak nokta anlamına gelmektedir. Bu çalışmalarda da, verilen bir maksimizasyon problemi için y noktası tarafından temsil edilen herhangi bir z noktası için yapılan temsil hatası $d(y, z) = \max_{i=1, \dots, m} (z_i - y_i)$ uzaklık ölçüsü kullanılarak hesaplanmaktadır. Diğer bir deyişle, sadece temsil edilen noktanın temsilcisinden iyi olduğu kriterler dikkate alınmaktadır.

Ceyhan vd. (2019), aynı uzaklık ve kapsama ölçüsünü kullanarak önerdikleri SBA (Subspace-Based Approach ve TDA (Territory Defining Algorithm) algoritmaları ile mevcut algoritmaları geliştirmişlerdir. Bu algoritmalar her adımda Lokman ve Köksalan (2013) tarafından önerilen yöntem ile çözüm uzayını alt uzaylara bölmektedirler. SBA her adımda her birden fazla altmodel çözerek bir sonraki temsilci noktayı bulmaktadır. Her alt uzayda en kötü temsil edilen baskın çözümü ürettikten sonra bunların arasından en kötü temsil hatasına sahip noktayı yeni temsilci olarak seçmektedir. TDA ise her adımda sadece en büyük hacimli alt uzay için bir altmodel çözmektedir. TDA her adımda en kötü temsil edilen noktayı bulmayı garantilemese de algoritma sonunda istenilen kapsama hatasını sağlamayı garantilemek için başka bir yöntem kullanmaktadır. Bu yöntemde, çözülen altmodellere eklenen bir kısıt ile her adımda mevcut noktalara izin verilen kapsama ölçüsünden daha yakın noktaların bulunması engellenmiştir. Ceyhan vd. (2019)

bu yöntem ile algoritma sonunda bütün baskın noktaların temsil hatasının en fazla istenilen kapsama ölçüsü kadar olduğunu göstermişlerdir.

Bütün bu mevcut yaklaşımlardaki temel amaç, verilen bir kapsama ölçüsünü olabilecek en küçük temsilci kümesi ile sağlamaktır. Algoritma sonunda en az sayıda temsilci üretebilmek için her adımda üretilen temsilcinin temsil ettiği nokta sayısının en fazla olması gerekmektedir. Bu doğrultuda, bu çalışmada ilk olarak temsilci olabilecek her bir baskın nokta için *bireysel temsil gücü* özelliği tanımlanmıştır.

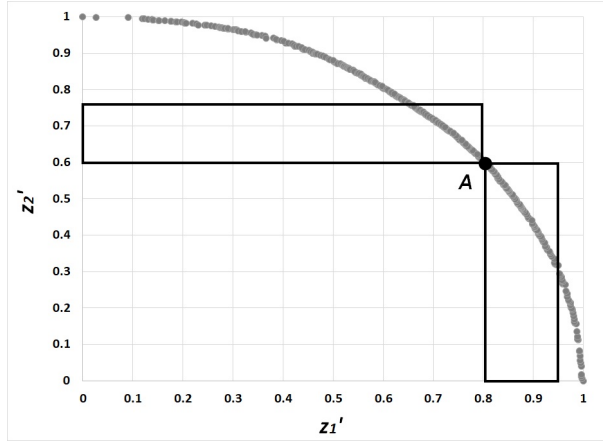
Tanım 11 (Bireysel Temsil Gücü): Bir baskın nokta y 'nin, verilen bir kapsama hatasına göre temsil edebildiği baskın nokta sayısı y 'nin bireysel temsil gücüdür ve $BTG(y)$ olarak gösterilir.

TSGA ve TSGA-II algoritmalarını geliştirirken temel amacımız her adımda bireysel temsil gücü en yüksek olan noktaları üretmektir. Bu doğrultuda, Bölüm 3.2'de yapılan yoğunluk analizleri ve Köksalan ve Lokman (2009) tarafından baskın nokta kümesinin şekline dair yapılan gözlemlerden faydalanılmıştır.

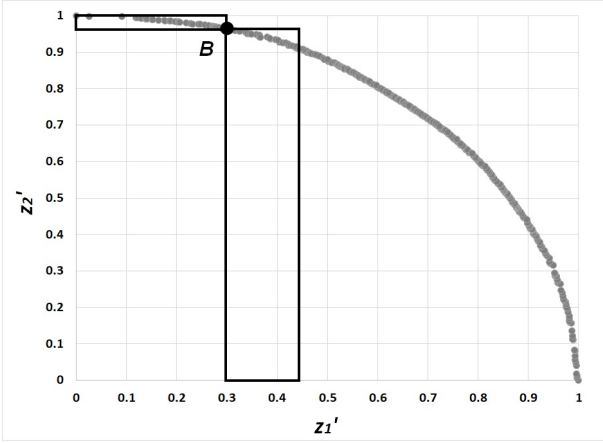
ÇATP'lerde baskın nokta kümesinin yoğunluk dağılımlarına yönelik yapılan analizlerde; bir problemdeki baskın noktaların yoğunluk değerlerinin problemin tipinden ve boyutundan bağımsız olarak merkeze doğru arttığı ve kenarlarda azalarak sığa yaklaştığı gözlemlenmiştir. Bu gözlemden yola çıkarak yoğun bölgelerde yer alan baskın noktaların bireysel temsil gücünün de fazla olacağı sonucu öngörülmüştür. Bunun temel sebebi, yoğun bölgedeki bir noktanın etrafında istenilen kapsama ölçüsüne göre temsil edebileceği nokta sayısının fazla olmasıdır.

Bu ilişki Şekil 13'te iki amaçlı ve 5652 baskın noktaya sahip bir sırt çantası problemi, ÇASP, üzerinde örneklendirilmiştir. Ölçeklendirilmiş baskın çözüm kümesindeki A ve B noktaları bireysel temsil gücü açısından kıyaslandığında, A noktasının B noktasına göre daha fazla baskın noktayı temsil ettiği görülmektedir. Şekilde her iki noktanın da istenilen 0,15 kapsama ölçüsüne göre, $\Delta = 0,15$, temsil ettiği bölgeler işaretlenmiştir. Bu durumun iki noktanın baskın nokta uzayındaki yerleri ile doğrudan ilişkisi vardır çünkü A noktası B noktasına göre daha yoğun bir bölgede (merkeze yakın) yer almaktadır.

Görüldüğü üzere yoğun bölgelerde yer alan baskın noktaların bireysel temsil güçleri de yüksek olmaktadır. Bu gözlemlere göre her adımda, mevcut üretilmiş çözümlerden en uzakta yer alan noktayı bulmak yerine; verilen kapsama hatasından uzak, ancak yoğun bölgelere yakın olan baskın noktayı bulan iki yeni yaklaşım (TSGA ve TSGA-II) geliştirilmiştir. Bu yaklaşımlarda birbirine daha yakın temsili baskın noktalar üretilirken; yoğunluğun fazla olduğu bölgelerde temsilciler seçilmesi planlanmaktadır.



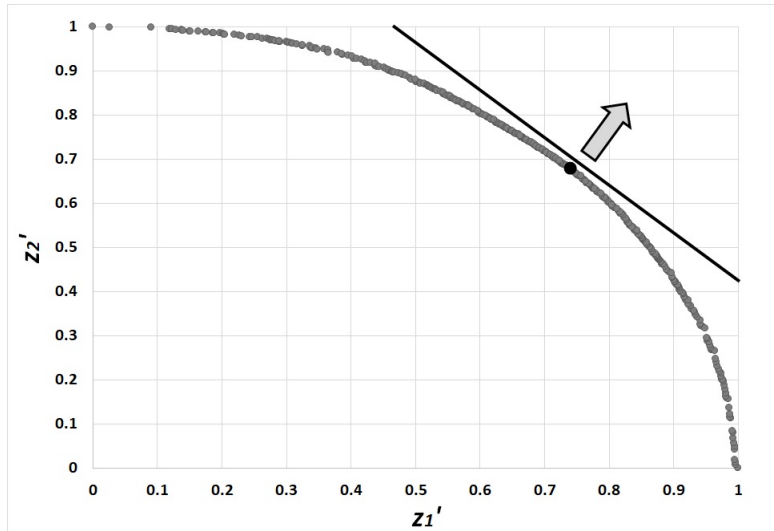
(a) Çok yoğun bölge - Yüksek BTG



(b) Az yoğun bölge - Düşük BTG

Şekil 13. Çözüm uzayındaki yoğunluk dağılımı ve bireysel temsil gücü ilişkisi

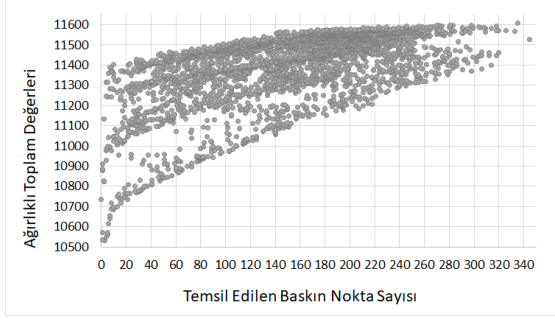
Yoğunluğun fazla olduğu bölgelerden temsilci seçilmesi için Köksalan ve Lokman (2009)'ın baskın nokta uzayının tipik şekillerine dair gözlemlerinden faydalanılmıştır. Bölüm 3.2'te de tartışıldığı gibi, Köksalan ve Lokman (2009) gerçek baskın nokta kümesini iyi temsil eden bir L_p yüzeyi tanımlamış ve yaptıkları deneyler ile ÇATP'lerde bazı tipik özellikleri gözlemlemiştir. Buna göre, tipik bir enazlama probleminde baskın nokta kümesinin şekli dışbükey iken ençoklama problemlerinde şekil içbükey olmaktadır. Bu doğrultuda, TSGA yaklaşımında her adımda merkeze yakın yoğunluğu yüksek bölgelerden baskın nokta üretmek için amaç fonksiyonu olarak kriterlerin pozitif ağırlıklandırılmış toplamı en iyilenmektedir. Bu ilişki Şekil 14'de iki amaçlı ve 431 baskın noktalı bir sırt çantası problemi üzerinde gösterilmiştir. Siyah doğru ile gösterildiği gibi iki amaç fonksiyonun ağırlıklı toplamları ençoklandığında baskın çözüm uzayının merkez bölgesinden bireysel temsil gücü yüksek bir baskın nokta üretilebilmektedir.



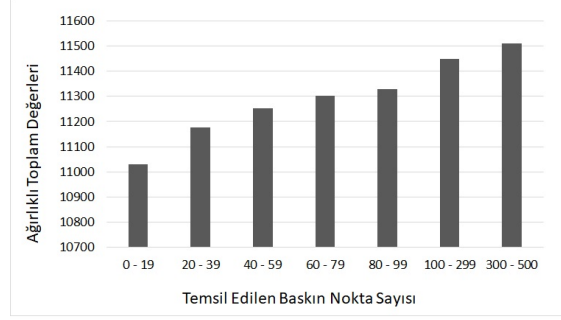
Şekil 14. Ağırlıklı-toplam modeliyle bireysel temsil gücü yüksek noktalar üretilmesi

Bu gözlemleri desteklemek için yapılan ek analizlerde verilen bir baskın nokta kümesindeki her bir noktanın verilen bir kapsama ölçüsüne göre bireysel temsil gücü ile o noktadaki

amaç fonksiyonunun değerlerinin ağırlıklı toplamları arasındaki ilişki incelenmiştir. Şekil 15a'da görüldüğü gibi baskın noktaların bireysel temsil güçleri ve amaç fonksiyon değerlerinin ağırlıklı toplamları arasında güçlü bir doğrusal ilişki vardır. Yine Şekil 15b'deki çubuk grafiğinden de gözlemlendiği gibi kriter değerlerini ağırlıklı toplam değerleri yüksek olan baskın noktalar ortalama daha çok nokta temsil edebilmektedir.



(a) Bireysel değerler



(b) Çubuk grafik

Şekil 15. Kriter değerlerinin ağırlıklı-toplamı ve bireysel temsil gücü ilişkisi

Burada önemli olan başka bir nokta ise ağırlıklı-toplam modellerinde kullanılacak kriter ağırlıklarının belirlenmesidir. Bu ağırlıklar verilen bir problemin baskın çözüm kümesinin yoğunluk dağılımına göre belirlenmektedir. Bölüm 3.2 kapsamındaki çalışmalarda görüldüğü gibi yoğunluğu yüksek olan bölgeler "yoğunluk merkezi" noktasının etrafında yer almaktadır. Bu sebeple, verilen bir problemin yoğunluk merkezine teğet olan yüzeyin katsayılarının normalleştirilmesiyle kriter ağırlıkları belirlenmiştir. Böylece, ağırlıklı-toplamın en iyileştirilmesiyle yoğunluk merkezine yakın baskın noktaların üretilmesi sağlanmıştır.

Bütün bu gözlemler ve analizler doğrultusunda geliştirilen TSGA algoritması, Ceyhan vd. (2019) tarafından önerilen modeli revize ederek, her adımda (P_n^{TSGA}) modelini çözmekte ve amaç fonksiyonlarının ağırlıklı toplam değeri en yüksek olan noktayı temsilci olarak seçmektedir. Ceyhan vd. (2019) tarafından önerilen TDA yaklaşımında olduğu gibi, (P_n^{TSGA}) modelindeki ($\alpha \geq \Delta$) kısıtı ile mevcut baskın noktalardan izin verilen kapsama ölçüsünden daha yakın baskın noktaların üretilmesi engellenmiştir. Böylece her adımda mevcut noktalar tarafından (kapsama ölçüsü açısından) temsil edilen tüm noktaların elendiği indirgenmiş bir baskın nokta uzayında ağırlıklı-toplam modeli çözülmektedir. Başka bir deyişle, her adımda indirgenmiş probleme ait destekli bir baskın nokta üretilmektedir. Fakat üretilen bu noktalar asıl problemde desteksiz baskın nokta olabilmektedir. Ayrıca, bu kısıtlar ile algoritma sonunda izin verilen kapsama ölçüsünün sağlanması garantilenmektedir.

(P_n^{TSGA}) :

$$\begin{aligned} \text{Maks} \quad & \sum_{i=1}^m w_i z_i(\mathbf{x}) \\ \text{Kısıtlar} \quad & z_i(\mathbf{x}) \geq (z_i^j(\mathbf{x}) + \alpha)t_{ij} - M_i(1 - t_{ij}) \quad \forall i, j \\ & \sum_{i=1}^m t_{ij} = 1 \quad \forall j \\ & \alpha \geq \Delta \\ & t_{ij} \in \{0, 1\} \\ & \mathbf{x} \in \mathbf{X} \\ & j = 1, \dots, n \\ & i = 1, \dots, m \end{aligned}$$

(P_n^{TSGA}) modelinde, t_{ij} ikili deęişkenleri ve eklenen kısıtlar ile yeni üretilecek baskın noktanın elimizdeki her bir j baskın çözümünden en az bir kriterde en az Δ kadar iyi olması garantilenmektedir. $t_{ij} = 0$ durumunda, ilgili kısıtın etkisiz hale gelmesi için $-M_i$ i . kriter için alt sınır olacak şekilde yeterince küçük bir sayı olarak tanımlanmıştır.

Son olarak, bu yaklaşım Dächert vd. (2017) tarafından önerilen yöntem ile çözüm uzayını alt uzaylara bölmektedir. Yapılan deneyler bu yöntemin mevcut yöntemlerden daha hızlı olduğunu göstermektedir. Bu nedenle, TSGA yaklaşımı Dächert vd. (2017) yöntemi ile (P_n^{TSGA}) modelini alt modellere bölünerek her adımda her bir alt uzay için bir model çözmektedir. Daha sonra üretilen bütün baskın noktalar arasından amaç fonksiyonlarının ağırlıklı toplam değeri en yüksek olan nokta temsilci olarak seçilmektedir. Alt modellere bölme yöntemi Lokman ve Köksalan (2013) ile aynı şekilde yapılmaktadır fakat Dächert vd. (2017) yöntemi her adımda alt uzayları tanımlayan kısıtları daha verimli bir yöntemle güncellemektedir.

TSGA yöntemi aşağıda aşama aşama özetlenmektedir.

Algorithm 4 TSGA

Aşama 1 Başlangıç. Algoritma $\mathbf{R} = \emptyset$ ve $n = 0$ değerleri ile başlatılır. Karar verici istenilen kapsama ölçüsü değerini, Δ , belirler.

Aşama 2 Yeni temsilci baskın noktanın bulunması.

Aşama 2.1. (P_n^{TSGA}) modeli farklı alt sınır vektörleri ile tanımlanan alt modellere bölünür.

Aşama 2.2. Her bir alt model çözülerek, olurlu problemlerin her biri için optimal noktalar bulunur. Alt problemlerin hiçbiri olurlu çözüm içermiyorsa, Aşama 3 ile algoritma durdurulur. En az bir çözüm olması durumunda Aşama 2.3 ile devam edilir.

Aşama 2.3. Olurlu alt model çözümlerinden amaç fonksiyonu değerlerinin ağırlıklı toplamı en büyük olan, (P_n^{TSGA}) probleminin optimal çözümü, \mathbf{z}^{n+1} , olarak belirlenir.

Aşama 2.4. $\mathbf{R} = \mathbf{R} \cup \{\mathbf{z}^{n+1}\}$, $n = n + 1$ olarak güncellenir. Aşama 2 tekrarlanır.

Aşama 3 Durma. Algoritma durdurulur. $\mathbf{R} = \{\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^n\}$ temsili baskın nokta altkümesini, n değeri ise temsili baskın noktaların sayısını gösterir.

Bölüm 4.4'te yer alan deney sonuçlarımıza göre, TSGA yaklaşımımız, karar verici tarafından verilen bir kapsama ölçüsünü sağlarken, literatürdeki diğer algoritalardan daha hızlı bir şekilde ve daha az sayıda temsili çözüm üretmenin yanı sıra optimal temsili çözüm sayısına da

olabildiğince yakın sonuçlar bulabilmektedir.

3.3.2 TSGA-II (Territory-Excluded Supported Generating Algorithm - II)

TSGA-II yaklaşımı, TSGA yaklaşımına benzer olarak verilen bir kapsama ölçüsü için her adımda amaç fonksiyonlarının ağırlıklı toplam değeri en yüksek olan baskın noktayı üretmektedir. Kriterlerin ağırlıkları da yine TSGA yaklaşımındaki gibi yoğunluk merkezine teğet olan yüzey ile belirlenmektedir. Bu yaklaşımın temel faydası, m kriterli bir problem için $(m - 1)$ boyutlu çözüm uzayında çalışma imkanı vermesidir.

Algoritmanın başında amaç fonksiyonlarından herhangi biri seçilir (Örneğin a kriteri). Her n adımında, üretilen mevcut noktalar tarafından henüz temsil edilemeyen baskın noktalar arasından a kriteri en iyi olan nokta bulunur, z^{IP^n} . Daha sonra, z^{IP^n} noktasını verilen kapsama ölçüsüne göre temsil edebilen baskın noktalar arasından ağırlıklı toplam değeri en yüksek olanı yeni temsilci olarak seçilir. Bu şekilde her adımda a kriteri açısından kötüleşen baskın noktalar üretilmekte olup, çözülen altmodellerdeki kapsama kısıtlarının a kriteri için yazılmasına gerek kalmamaktadır. Aynı zamanda, altmodelin amaç fonksiyonunda a kriteri dışındaki $(m - 1)$ kriterin ağırlıklı-toplamları en iyileştirilmektedir. Bu sayede, verilen bir problemin boyutu, kriter sayısı açısından 1 azaltılmaktadır.

Bu şekilde, TSGA-II her adımda iki model (P_{n1}^{TSGAII}) ve (P_{n2}^{TSGAII}) çözümlenerek iki baskın nokta üretilmektedir. Üretilen ilk baskın nokta (z^{IP^n}) ikinci modele girdi olarak kullanılmaktadır.

(P_{n1}^{TSGAII}):

$$\begin{aligned} \text{Maks } & w_a z_a(\mathbf{x}) + \epsilon \sum_{i=1, i \neq a}^m w_i z_i(\mathbf{x}) \\ \text{Kısıtlar } & z_i(\mathbf{x}) \geq (z_i^j(\mathbf{x}) + \alpha) t_{ij} - M_i(1 - t_{ij}) \quad \forall i \neq a, \forall j \\ & \sum_{i=1, i \neq a}^m t_{ij} = 1 \quad \forall j \\ & \alpha \geq \Delta \\ & t_{ij} \in \{0, 1\} \\ & \mathbf{x} \in \mathbf{X} \\ & j = 1, \dots, n \\ & i = 1, \dots, m \end{aligned}$$

(P_{n2}^{TSGAII}) :

$$\begin{aligned} \text{Maks} \quad & \sum_{i=1, i \neq a}^m w_i z_i(\mathbf{x}) + \epsilon w_a z_a(\mathbf{x}) \\ \text{Kısıtlar} \quad & z_i(\mathbf{x}) + \alpha \geq z_i^{IP^n} \quad \forall i \\ & z_i(\mathbf{x}) \geq (z_i^j(\mathbf{x}) + \alpha) t_{ij} - M_i(1 - t_{ij}) \quad \forall i \neq a, \forall j \\ & \sum_{i=1, i \neq a}^m t_{ij} = 1 \quad \forall j \\ & \alpha \geq \Delta \\ & t_{ij} \in \{0, 1\} \\ & \mathbf{x} \in \mathbf{X} \\ & j = 1, \dots, n \\ & i = 1, \dots, m \end{aligned}$$

Algorithm 5 TSGA-II

Aşama 1 Başlangıç. Algoritma $\mathbf{R} = \emptyset$ ve $n = 0$ değerleri ile başlatılır. Karar verici istenilen kapsama ölçüsü değerini, Δ , belirler.

Aşama 2 z^{IP^n} baskın noktasının bulunması.

Aşama 2.1. (P_{n1}^{TSGAII}) modeli farklı alt sınır vektörleri ile tanımlanan alt modellere bölünür.

Aşama 2.2. Her bir alt model çözülerek, olurlu problemlerin her biri için optimal noktalar bulunur. Alt problemlerin hiçbiri olurlu çözüm içermiyorsa, Aşama 4 ile algoritma durdurulur. En az bir çözüm olması durumunda Aşama 2.3 ile devam edilir.

Aşama 2.3. Olurlu alt model çözümlerinden a amaç fonksiyonu değeri en büyük olan, (P_{n1}^{TSGAII}) probleminin optimal çözümü, z^{IP^n} , olarak belirlenir.

Aşama 3 Yeni temsilci baskın noktanın bulunması.

Aşama 3.1. (P_{n2}^{TSGAII}) modeli farklı alt sınır vektörleri ile tanımlanan alt modellere bölünür.

Aşama 3.2. Her bir alt model çözülerek, olurlu problemlerin her biri için optimal noktalar bulunur. Alt problemlerin hiçbiri olurlu çözüm içermiyorsa, z^{IP^n} baskın noktası, (P_{n2}^{TSGAII}) probleminin optimal çözümü, z^{n+1} , olarak belirlenir ve Aşama 3.4 ile devam edilir. En az bir çözüm olması durumunda ise Aşama 3.3 ile devam edilir.

Aşama 3.3. Olurlu alt model çözümlerinden amaç fonksiyonu değerlerinin (a kriteri hariç) ağırlıklı toplamı en büyük olan, (P_{n2}^{TSGAII}) probleminin optimal çözümü, z^{n+1} , olarak belirlenir.

Aşama 3.4. $\mathbf{R} = \mathbf{R} \cup \{z^{n+1}\}$, $n = n + 1$. Aşama 2 tekrarlanır.

Aşama 4 Durma. Algoritma durdurulur. $\mathbf{R} = \{z^1, z^2, \dots, z^n\}$ temsili baskın nokta altkümesini, n değeri ise temsili baskın noktaların sayısını gösterir.

TSGA-II'nin başka bir önemli özelliği ise iki kriterli karışık tamsayı programlama modellerinde, verilen bir kapsama ölçüsünü optimal temsili çözüm sayısı ile sağlamasıdır. Daha önce de belirtildiği gibi TSGA ve TSGA-II yaklaşımlarındaki temel motivasyonumuz her adımda bireysel temsil gücü yüksek noktaları üretmektir. İki kriterli problemlerde, TSGA-II yaklaşımının her adımda bireysel temsil gücü maksimum olan baskın noktayı üreterek toplam temsilci sayısını enazladığı

kanıtlanmıştır (bakınız Dođan, 2018). Ayrıca, TSGA-II yaklaşımın başında seçilen a kriterinin üretilen temsilci sayısına etkisini incelemek için yapılan ek deneylerde, rastgele seçilen bir kriterin çözüm kalitesi üzerinde belirgin bir etkisinin olmadığı gözlenmiştir.

TSGA-II yaklaşımı Dächert vd. (2017) yöntemi ile (P_{n1}^{TSGAII}) ve (P_{n2}^{TSGAII}) modellerini altmodellere bölerek çözmektedir. Fakat her bir altuzay için ikişer model çözülmesi algoritmanın çözüm süresini uzatmaktadır. Her bir altuzay için (P_{n1}^{TSGAII}) modelini bir kere çözmek yeterli olsa da, z^{IP^n} girdisi her adımda değiştiđi için (P_{n2}^{TSGAII}) modelini her adımda her bir altuzay için yeniden çözmek gerekmektedir. Bu durum, problem boyutu arttıkça TSGA-II yaklaşımını verimsiz bir hale getirirse de, deneylerimizde üretilen temsilci sayısı açısından değerlendirildiđinde mevcut algoritmalarından daha iyi olduđu gösterilmiştir. Kısacası, TSGA-II çözüm süresi açısından mevcut algoritmalarından daha kötü performans sergilese de çözüm kalitesi açısından önemli iyileştirmeler sağlamaktadır. Özellikle iki kriterli problemlerde, TSGA-II, istenilen bir kapsama ölçüsünü sağlayan, en küçük temsili kümeyi bulmaktadır. Yapılan deney sonuçları Bölüm 4.4'te sunulmaktadır.

4 BULGULAR

ÇATP Problemleri için Bölüm 3'te yapılan incelemeler ve geliştirilen yöntemler kullanılarak tasarlanan yazılımlar, araçlar ve yapılan uygulamalar bulguları ile beraber bu bölümde sunulmaktadır.

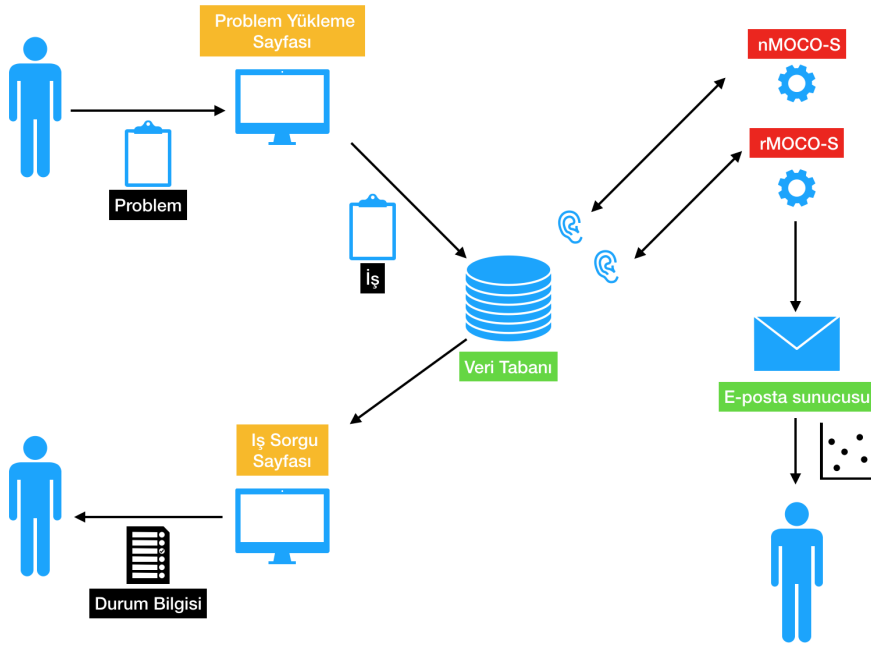
4.1 Geliştirilen Algoritmaların ve Görselleştirme Araçlarının İnternet Tabanlı Bir Uygulama Olarak Sunulması

Geliştirilen algoritmaların ve araçların internet tabanlı bir uygulama olarak kullanıcılara sunulması, yapılan araştırmaların diğer araştırmacılar tarafından kolaylıkla kullanılabilmesi için önemli olmaktadır. Ayrıca, değişik tamsayı programlama problemleri için oluşturulmuş test örneklerinin ve bu örnekler üzerinde yapılan koşumların sonuçlarının yayımlanarak diğer araştırmacıların erişimine açık bir test kütüphanesi oluşturulması, gelecek dönemlerde bu konu üzerinde yapılacak olası çalışmalar için bir zemin hazırlamaktadır. Geliştirdiğimiz web uygulaması ve çözücü platformu sayesinde, bir kullanıcı karşılaştığı bir çok amaçlı tamsayı problemini kolayca geliştirdiğimiz çözücülere aktarabilecek ve elde edilen sonuçlara erişip analiz edebilecektir.

4.1.1 Uygulamalar

Geliştirmiş olduğumuz yazılım, <http://www.onlinemoco.com/MOIP/> adresinde sunulmuş olup dört ana uygulamadan oluşacak şekilde tasarlanmıştır. Bu uygulamaların, aşağıda belirtilen amaçlar için kullanılması amaçlanmıştır.

1. **nMOCO-S**: Bu uygulama, verilen bir ÇATP için eğer bir süre limiti yok ise, tüm baskın çözümleri (n) üretir. Eğer bir süre limiti var ise, bu süre limiti içinde elde edilen tüm baskın çözümler problem sahibine iletilir.
2. **rMOCO-S**: Bu uygulamada amaç, tüm baskın çözümler yerine, baskın çözüm kümesini belirli kalite ölçüleri bakımından iyi temsil edecek az sayıda (r) temsili baskın çözüm üretmektir. Bu uygulama altında iki çözücü seçilebilmektedir: SBA (subset based algorithm) ve TDA (territory defining algorithm). İlk çözücü, kullanıcının belirlediği çözüm sayısında ulaşıncaya kadar veya süre limiti dolana kadar baskın çözümler üretmektedir. İkinci çözücü ise, kullanıcının belirlediği kapsama değerine ulaşıncaya kadar veya süre limiti dolana kadar baskın çözümler üretmektedir.
3. **libMOCO-S**: Bu uygulamanın, dijital bir ÇATP test kütüphanesi (lib) olarak hizmet vermesi amaçlanmıştır. Farklı tamsayı veya bileşi optimizasyonu problemlerinde değişik özelliklerde oluşturulmuş rassal örnekleri ve kullanıcılardan gelen problemleri içermesi planlanmaktadır. Bu örnekler, bazı özellikleri ve baskın çözüm kümeleri ile birlikte bir veri tabanında saklanarak, kullanıcılar tarafından görüntülenebilecektir. Uygulamanın mevcut durumda, 4 amaçlı sırt çantası problem örnekleri ve 5 amaçlı atama problemi örnekleri sergilenmektedir.



Şekil 16. nMOCO-s ve rMOCO-S çözücülerinin bulut servis olarak kullanım şeması

Gelistirdiğimiz uygulamada, kullanıcılar tarafından girilen problemler birer iş olarak tanımlanmış, bu işlerin yönetilmesi için bir kuyruk yapısı oluşturulmuş, çözücülerin iş kuyruğu ile iletişim protokolü ve sonuçların kullanıcıya gönderilmesi için e-posta sunucusu geliştirilmiştir. Ayrıca, ilgili veri akışlarının sağlanması için bir veri tabanı oluşturulmuştur. Şekil 16'de sunucu tarafındaki çözücüler, kullanıcı ara yüzleri, e-posta sunucusu ve veri tabanı arasındaki etkileşim sergilenmektedir.

Bu tasarımda, “problem”, “iş” ve “durum bilgisi” nesnelerinin özellikleri ile kuyruk ve çözücülerin çalışma prensibi aşağıdaki gibidir:

1. **Problem:** Problem nesnesi, kullanıcı bilgileri, problem verisi veya parametreleri, ve kullanılmak istenen çözücü türünden oluşmaktadır.
2. **İş:** Kullanıcı tarafından girilen problem nesnesi, bu probleme atanan bir iş numarası, problemin girildiği zaman, iş durum bilgisi, iş bitiş zamanı ve çözüm süresi değerlerinden oluşmaktadır. İş numarası, girilen her iş için farklı olacak şekilde veri tabanı tarafından otomatik olarak atanır. İş durum bilgisinin başlangıç değeri ise “*TO – DO*” olarak belirlenir. Girilen iş, kuyruğa eklenir ve kullanıcıya iş numarası bildirilir.
3. **Durum Bilgisi:** Durum bilgisi nesnesi dört değer alabilmektedir. Bu dört değer ve açıklamaları aşağıdaki gibidir:
 - (a) **TO DO:** Girilen işin henüz çözücü tarafından işleme alınmadığını belirtir. Bu durumda, kullanıcıya iş sorgu sayfasında işin kuyruk sırası ve tahmini bekleme süresi bilgisi verilir.

- (b) **IN PROGRESS:** Girilen işin çözücü tarafından işlenmekte olduğunu gösterir.
- (c) **FINISHED FAIL:** Girilen işin çözücü tarafından ele alındığını ancak bir sebepten ötürü işin sonuçlandırılmadığını belirtir. İlgili hata kayıtları, kullanıcıya gönderilecek dosyada bulunur.
- (d) **FINISHED SUCCESS:** Girilen işin çözücü tarafından ele alındığını ve başarılı bir şekilde sonuçlandırıldığını belirtir. Elde edilen sonuç dosyası, e-posta sunucusu tarafından kullanıcıya gönderilir.

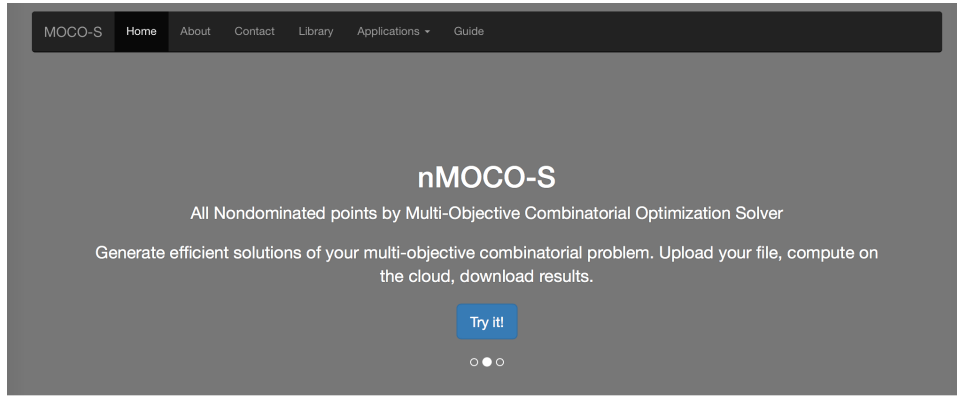
4. **Kuyruk:** “İlk giren ilk çıkar” prensibine göre çalışır.

5. **Çözücüler:** nMOCO-S ve rMOCO-S çözücüleri, uygulama yöneticisi tarafından belirlenen aralık parametresine (2 dakika) göre veri tabanını kontrol ederek, veri tabanında “TO DO” durumunda en az bir iş bulunması durumunda o iş ile ilgili problemi çözmeye başlarlar. Çözücüler bir iş için en fazla uygulama yöneticisi tarafından belirlenecek süre limiti kadar çalışabilirler. Bu süre limitine vuran işler için, sonuç dosyasında ilgili bilgilendirme yapılır. Bir iş bitimi sonrasında, eğer veri tabanından “TO DO” durumunda başka bir iş var ise, çözücü ara vermeden o işi ele alır. Veri tabanında, bu durumda başka bir iş bulunmadığında ise, çözücü belirlenen aralıkta düzenli bir şekilde veri tabanını kontrol eder.

Yazılımın ön yüzlerinin geliştirilmesinde Java Server Pages, JavaScript ve Html dilleri kullanılmıştır. Ayrıca, jQuery kütüphanesinden yararlanılmıştır. Sunucu tarafında ise Servlet teknolojisi yardımı ile çözücüler ile iletişim kurulmaktadır. Çözücüler, daha iyi performans göstermeleri amacı ile C++ dilinde kodlanmıştır. Web sunucusu olarak Apache Tomcat, veri tabanı sunucusu olarak da Derby kullanılmaktadır. Sunucu olarak, Windows Server 2012 işletim sistemine sahip Intel E5 işlemcili 32 GB RAM belleğe sahip HP marka bilgisayar kullanılmaktadır.

4.1.2 Ekranlar

1. **Ana sayfa:** Bu sayfada kullanıcılara, uygulamasının ana modülleri hakkında özet bilgi sunulmaktadır (17).
2. **Hakkında sayfası:** Bu sayfada, proje ve proje kapsamında geliştirilen yazılım hakkında kullanıcılara genel bilgiler, proje özeti ile beraber sunulmaktadır. Ayrıca uluslararası konferanslarda sunulan çalışmalar da ek bilgi için yer almaktadır.
3. **İletişim bilgileri sayfası:** Bu sayfada, projenin yürüttüğü yer ve yazışma bilgileri sunulmaktadır.
4. **Problem yükleme sayfası:** Kullanıcı bu ekran üzerinden, problem dosyasını yükler, kullanmak istediği çözücü seçer ve ilgili parametre değerlerini ve kullanıcı iletişim bilgilerini girerek problemi sunucuya gönderir, Şekil 18.
5. **Çözüm durumu sorgulama sayfası:** Kullanıcı problemi yükledikten sonra, bir iş numarası elde eder. Girilen e-posta adresi ve bu iş numarasını kullanarak, kullanıcı probleminin



Murat Köksalan

View »



Banu Lokman

View »



Gökhan Ceyhan

View »

Şekil 17. Ana sayfa görüntüsü

çözücü kuyruğundaki sıra numarasını görebilir ve çözüm durumu hakkında bilgi elde edebilir, Şekil 19.

6. **Kütüphane sayfası:** Örnek problemlerin girdi dosyaları ve çözümlerimiz tarafından elde edilen sonuçları sunulmaktadır, Şekil 20. Bu sayfada hem .lp formatında hem de .txt formatında girdi dosyaları örnekleri ve bu örnekler ile sunucu üzerinde çalıştırılan çözümlerin elde ettikleri sonuç dosyaları sunulmaktadır. Bu sonuç dosyaları içerisinde çözüm süresi, bulunan çözüm sayısı, kalite ölçüsü gibi değerler raporlanmaktadır. Böylece, araştırmacıların bu örnekler üzerinde çalışmalarını test edebilmelerine yardımcı olmak amaçlanmaktadır.

7. **Kullanıcı kılavuz sayfası:** Bu sayfada, çözümler hakkında bilgi verilmekte, hangi çözümlerin hangi kullanım amacına uygun olduğu anlatılmakta, istenilen girdi ve elde edilen çıktı dosyalarının formatları belirtilmekte ve kullanılan parametre değerleri verilmektedir.

Uygulamanın kaynak koduna: <https://github.com/gokhanceyhan/MOCOS> adresinden, çözümlerin kaynak koduna ise https://github.com/gokhanceyhan/MOIP_Solvers adresinden ulaşılabilir.

4.2 Geliştirilen Algoritmalar Tarafından Bulunan Çözümler için Görselleştirme Araçlarının Geliştirilmesi

Bu bölümde, üretilen çözüm kümesinin ve bu çözüm kümesine ait kalite ölçülerinin kullanıcıya raporlanması için geliştirilen görselleştirme araçları aktarılmaktadır. Çok amaçlı optimizasyon problemlerinde, bulunan baskın çözümler arasındaki ödünleşmenin ve baskın çözüm uzayının özelliklerinin problem sahibi tarafından iyi kavranabilmesi için görselleştirme araçlarının kullanımı gerekmektedir. Amaç fonksiyonu sayısının iki olduğu problemlerde baskın çözümlerin çözüm

MOCO-S Home About Contact Library Solvers Guide

SOLVER PARAMETERS

Solver Type
nMOCO-S: Generate all nondominated points

Time Limit in seconds
600

Email
✕ Email

I agree with the [terms and conditions](#)

Upload

INPUT PARAMETERS

Input Type
Model File (.lp)

Number of Objectives
0

File Input
Choose File no file selected

Şekil 18. Problem dosyası yükleme ekranı


MOCO-S Home About Contact Library Solvers Guide

✕ Email 🔍 Job ID **Search**

Search Results

| Job Id | Issuer | Job Creation Time | Seq. Number | Job Status | Job Completion Time | Processing Time (secs) |
|--------|--------------------------|-------------------------|-------------|------------------|-------------------------|------------------------|
| 1457 | gokhanceyhan01@gmail.com | 2018-10-08 22:33:47.269 | 0 | FINISHED_SUCCESS | 2018-10-08 22:34:30.113 | 33.0 |

Şekil 19. Çözüm sorgulama ekranı



Library for Multi Objective Combinatorial Optimization

[\[Model & Data Format\]](#)
Knapsack Problem 🔍

| Instance Id | Number of Objectives | Number of Knapsacks | Number of Items | Number of Nondominated Points | Input File | Output File |
|-------------|----------------------|---------------------|-----------------|-------------------------------|----------------------------|-----------------------------|
| 1 | 4 | 1 | 25 | 87 | Input File | Output File |
| 2 | 4 | 1 | 25 | 197 | Input File | Output File |
| 3 | 4 | 1 | 25 | 232 | Input File | Output File |
| 4 | 4 | 1 | 25 | 87 | Input File | Output File |
| 5 | 4 | 1 | 25 | 163 | Input File | Output File |
| 6 | 4 | 1 | 50 | 1512 | Input File | Output File |
| 7 | 4 | 1 | 50 | 1439 | Input File | Output File |
| 8 | 4 | 1 | 50 | 966 | Input File | Output File |
| 9 | 4 | 1 | 50 | 1879 | Input File | Output File |
| 10 | 4 | 1 | 50 | 1616 | Input File | Output File |

Şekil 20. Kütüphane sayfası

uzayındaki dağılımı basit bir dağılım grafiği ile kullanıcıya etkin bir şekilde aktarılabilmektedir. Ancak, amaç sayısının üç veya daha fazla olduğu problemlerde görselleştirme araçları daha fazla önem kazanmaktadır.

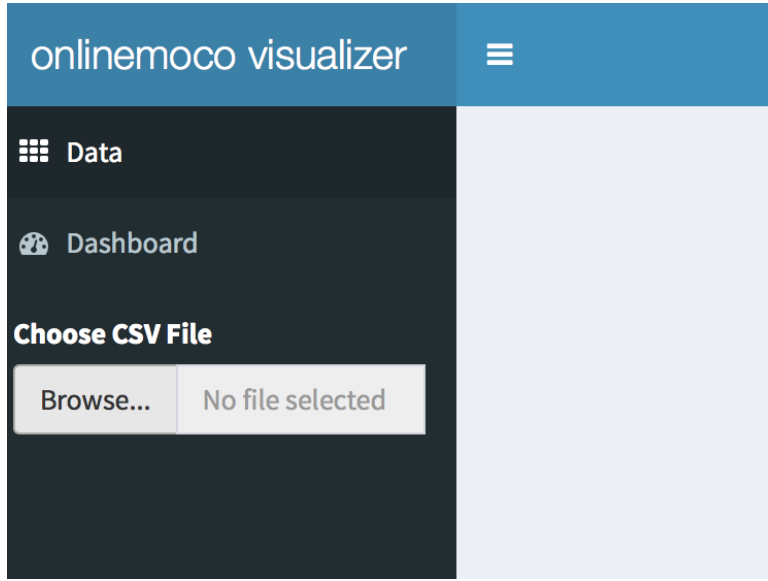
Çok amaçlı karar verme problemlerinde görselleştirme araçlarının, karar vericinin çözüm uzayı algısı ve en çok tercih ettiği çözümün seçimi üzerine etkileri değişik çalışmalarda incelenmiştir (Lotov ve Miettinen, 2008; Gettinger vd., 2013; Korhonen ve Wallenius, 2008). Literatürdeki çalışmalar görselleştirme araçlarını özellikle kullanıcı etkileşimli yöntemler üzerinden incelerken, bizim amacımız tüm baskın çözümlerin veya temsili baskın çözüm kümesinin çözüm uzayındaki yerleşiminin etkin bir şekilde kullanıcı tarafından algılanmasını sağlamak olmuştur. Aynı zamanda, kullanıcıların baskın çözümlerin çözüm uzayındaki yoğunluk dağılımını kavramaları ve tercih bölgelerindeki ödünleşme seviyesi hakkında bilgilenmeleri amaçlanmıştır.

Baskın çözüm kümelerinin kullanıcılara sunulması amacı ile bir web uygulaması yazılmış ve bu uygulama üzerinde iki tür interaktif grafik geliştirilmiştir. Tasarım aşamasında, grafiklerin aşağıdaki fonksiyonları yerine getirebilmesi amaçlanmıştır:

1. Üçten fazla kriterli problemler için de kullanılabilmesi
2. Kriter değerleri üzerinde kullanıcı tarafından limitler konulduğunda, grafiklerin sadece bu limitler dahilindeki baskın noktaları göstermesi
3. Üç kriterli problemlerde, kullanıcının üç boyutlu düzlemde farklı açılardan baskın nokta kümesini görüntüleyebilmesi ve ilgilendiği bölgelerin yakınlaştırılması
4. Kullanıcının seçtiği baskın noktaların grafikler üzerindeki yerlerinin gösterilmesi

Belirtilen gereksinimlerin karşılamak adına en fazla üç kriterli problemler için kullanılacak bir dağılım grafiği ve her sayıda kriter için kullanılacak bir paralel koordinatlar grafiği geliştirilmiştir. Görselleştirme uygulaması R dilinde yazılmış olup, “shiny” ve “plotly” kütüphaneleri kullanılmıştır. Aşağıdaki örnekte, görselleştirme uygulamamızın örnek bir problemin baskın çözüm kümesini görselleştirmek için nasıl kullanılacağı gösterilmektedir.

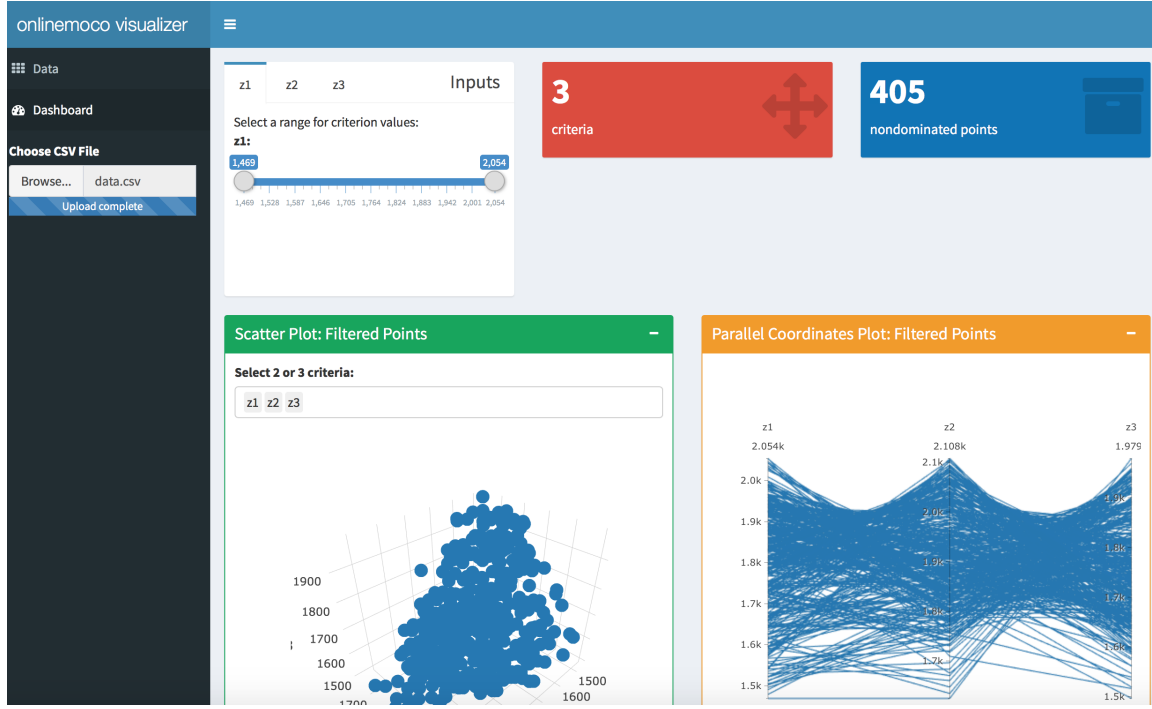
1. Kullanıcı ilk olarak, web üzerindeki çözümlerimiz sonucu olarak kendisine gelen e-postadaki sonuç dosyasını görselleştirme uygulamamızdaki ilgili alandan yükler, Şekil 21.
2. Yükleme sonrasında, “Dashboard” sayfasındaki grafikler üzerinde sonuç dosyasında yer alan tüm baskın noktalar gösterilir, Şekil 22.
3. Kullanıcı, sol üst köşedeki filtreleme aracını kullanarak görüntülemek istediği baskın noktaların kriter değerleri üzerinde alt ve üst sınırlar belirleyebilir. Filtreleme aracında her bir kriter için izin verilen maksimum değer ideal noktanın ilgili kriter değeri, minimum değer ise nadir noktanın ilgili kriter değeridir. Örneğin, z1 kriteri üzerinde [1657, 1788], z2 kriteri üzerinde [1727, 1969] ve z3 kriteri üzerinde [1657, 1864] aralıklarını seçtiğimizde, grafikler sadece bu kısıtları sağlayan 18 baskın noktayı göstermektedirler, Şekil 23.



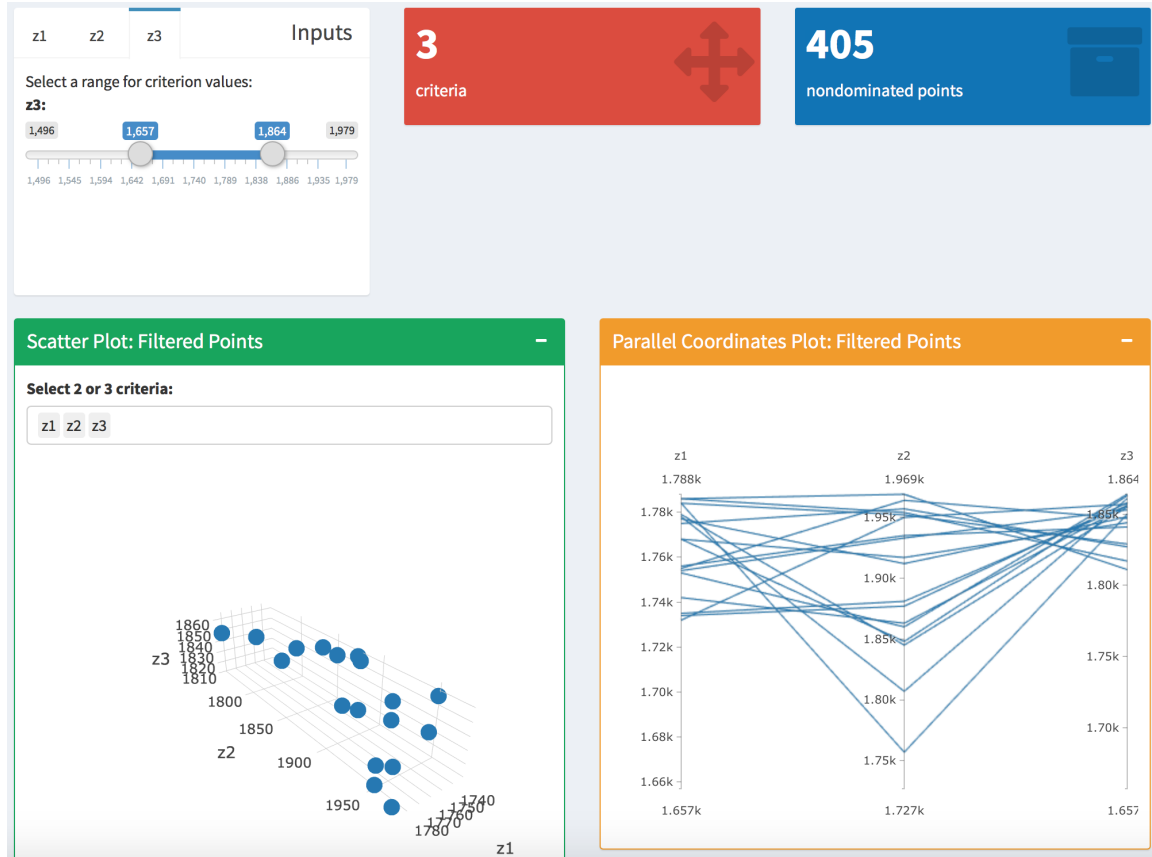
Şekil 21. Baskın noktaların bulunduğu dosyanın yüklenmesi

4. Kullanıcı dağılım grafiği üzerine imlecini getirdiğinde, noktaların kriter değerleri görüntülenmekte, istenilen noktalara doğru odaklanılabilmekte ve grafiğe farklı açılardan bakılabilmektedir. Aynı zamanda, baskın noktaların seçilecek iki boyutlu düzlemde gösterimi de yapılabilmektedir, Şekil 24, 25.
5. Filtreleme sonucunda elde edilen baskın nokta alt kümesi, bir tabloda kullanıcıya gösterilmekte, kullanıcının imleci ile seçtiği noktalar grafikler üzerinde gösterilmektedir, Şekil 26 - 28.

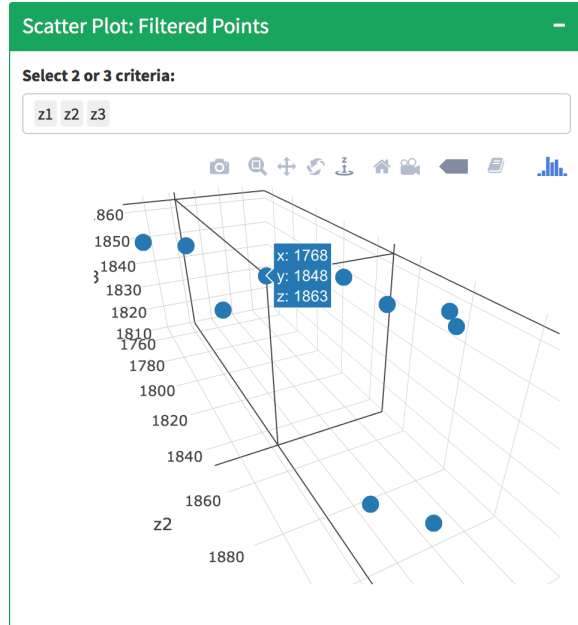
Geliştirdiğimiz bu uygulamanın kaynak koduna <https://github.com/gokhanceyhan/vMOCO> bağlantısından ulaşılabilir.



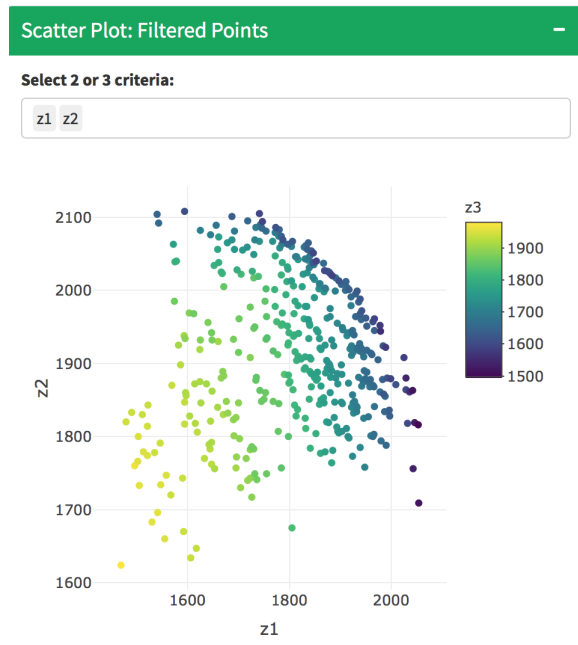
Şekil 22. Ana sayfa tasarımı



Şekil 23. Belirli bir bölgedeki noktaların filtrelenmesi



Şekil 24. Kriter uzayının interaktif bir şekilde taranması



Şekil 25. Baskın noktaların z1-z2 düzlemine projeksiyonu

Filtered Points

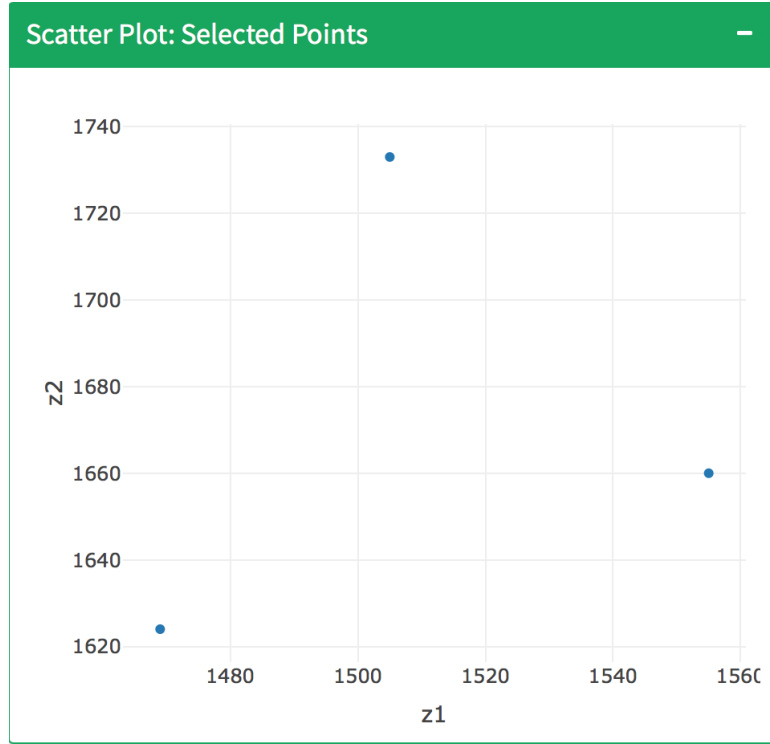
Show 10 entries Search:

| | z1 | z2 | z3 |
|----|------|------|------|
| 1 | 1469 | 1624 | 1979 |
| 2 | 1496 | 1760 | 1969 |
| 3 | 1505 | 1733 | 1959 |
| 4 | 1530 | 1683 | 1959 |
| 5 | 1541 | 1696 | 1958 |
| 6 | 1502 | 1766 | 1956 |
| 7 | 1513 | 1779 | 1955 |
| 8 | 1555 | 1660 | 1954 |
| 9 | 1521 | 1774 | 1953 |
| 10 | 1547 | 1734 | 1953 |

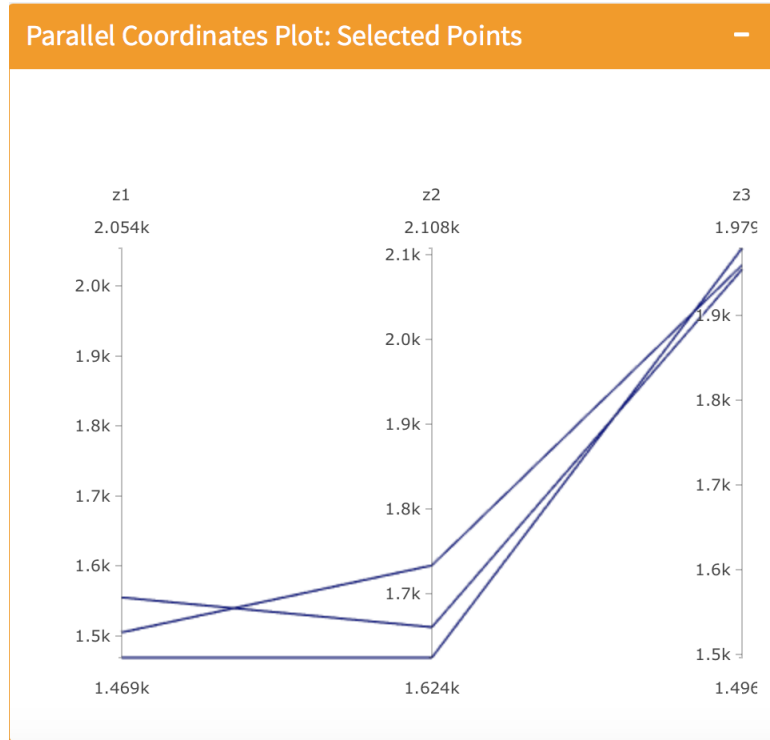
Showing 1 to 10 of 405 entries

Previous 1 2 3 4 5 ... 41 Next

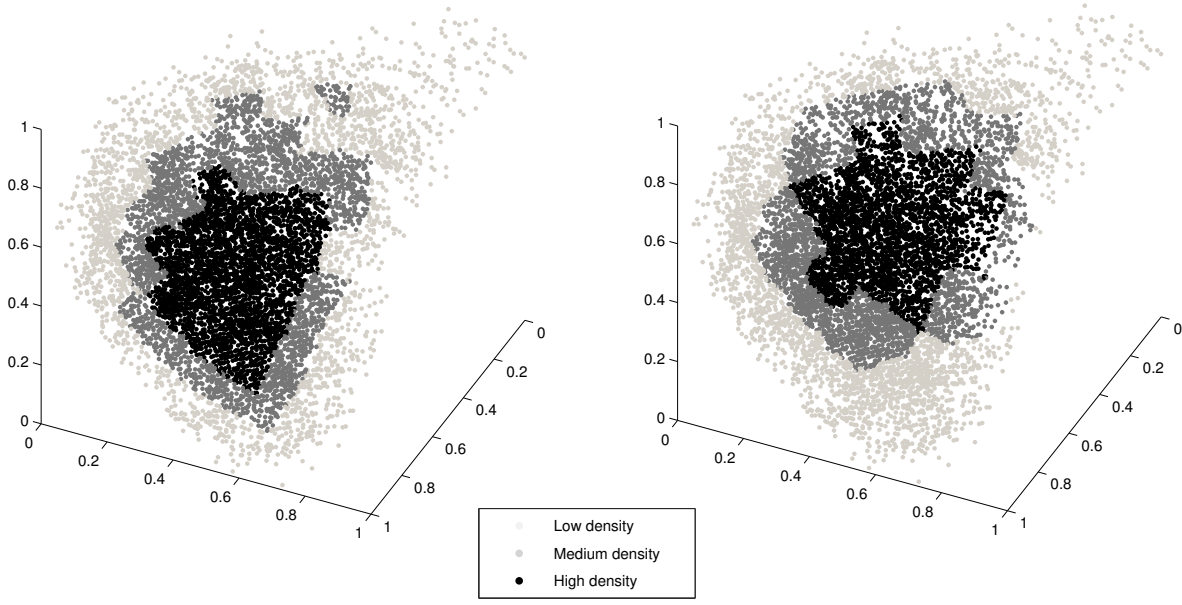
Şekil 26. Seçilen baskın noktaların filtreleme ile görselleştirme araçlarında karşılaştırılması



Şekil 27. Seçilen baskın noktaların dağılım grafiği üzerinde görselleştirme araçları ile karşılaştırılması



Şekil 28. Seçilen baskın noktaların paralel koordinatlar grafiği üzerinde görselleştirme araçlarında karşılaştırılması



a. True categories based on PWA density values

b. Estimated categories based on DBA

Şekil 29. 3-amaçlı 100-parçalı ÇASP için baskın noktaların yoğunluk kategorileri ($N = 6500$; $C = 3$; $h = 0, 2$; $e = 0, 1$)

4.3 DBA ile Deneyler

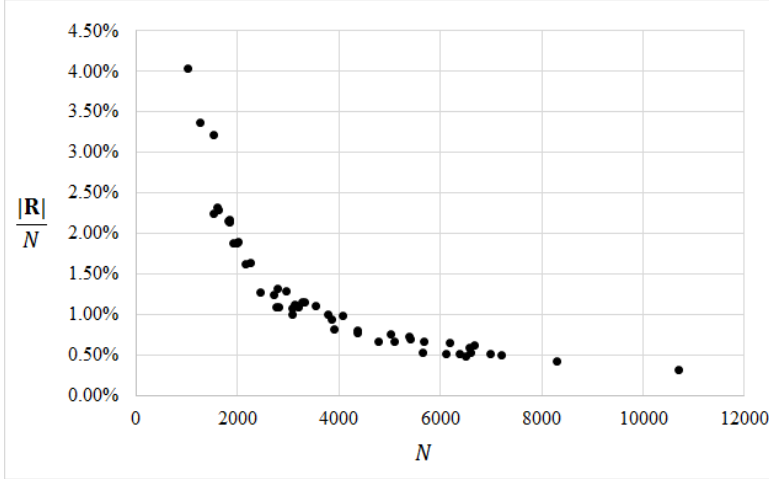
Bölüm 3.2'deki analizler sonucunda tasarlanan, baskın nokta kümesinin dağılım özelliklerini dikkate alarak temsili altküme üreten, DBA'nın performansını değerlendirmek amacıyla farklı ÇATP'lerde deneyler yapılmış, bulgular bu bölümde sunulmuştur. Bu deneylerde, baskın nokta kümesinin bilinmediği varsayılmış algoritmanın sadece performansını değerlendirilirken gerçek baskın nokta kümesi kullanılmıştır. Karar verici tarafından istenilen λ and $\alpha(\lambda)$ kapsama ölçüsüne göre temsili baskın nokta kümesi üretmek amaçlanmıştır. Değerlendirme amaçlı, üretilen küme için gerçek baskın nokta kümesi ve gerçek kategoriler kullanılarak gerçek ağırlıklı kapsama ölçüsü, $\alpha_a(\lambda)$, hesaplanmıştır. Şekil 29'da da gösterildiği gibi tahmin edilen kategoriler gerçek kategorilerden farklı olabileceği için ayrıca tahmin edilen kategorilere göre de ağırlıklı kapsama ölçüsü $\alpha_p(\lambda)$ bulunmuştur.

Ayrıca, temsili hipotetik nokta üretilmesi kısmında algoritmamızın performansını ölçmek için ise, gerçek kategorilerin algoritma boyunca bilindiği varsayımı altında DBA tekrar çalıştırılmıştır ve bu durumda elde edilen altkümenin ağırlıklı kapsama ölçüsü $\alpha_t(\lambda)$ karşılaştırma amaçlı raporlanmıştır. Tablo 5, baskın nokta sayısı [1014, 10701] aralığında olan 3-amaçlı ÇAAP, ÇASP, ÇAKPP'leri için karar verici tarafından seçilen $\lambda = (1, 00; 0, 75; 0, 5)$ and $\alpha(\lambda) = 0, 10$ hesaplanan bu ağırlıklı kapsama ölçülerini sunmaktadır.

Tablo 5. DBA için sonuçlar* ($\alpha(\lambda) = 0, 10$, $\lambda = (1.00, 0.75, 0.5)$)

| | N | $ R $ | $\alpha_a(\lambda)$ | $\alpha_p(\lambda)$ | $\alpha_t(\lambda)$ |
|----------------|----------|-------|---------------------|---------------------|---------------------|
| minimum | 1014,00 | 30,00 | 0,0807 | 0,0775 | 0,0759 |
| maksimum | 10701,00 | 49,00 | 0,1697 | 0,1697 | 0,1697 |
| ortalama | 3973,78 | 36,27 | 0,1056 | 0,1051 | 0,1031 |
| standart sapma | 2115,02 | 3,59 | 0,0188 | 0,0194 | 0,0165 |

*51 problem üzerinden hesaplanmıştır.



Şekil 30. Üretilen temsilci kümesinin tüm baskın nokta kümesine göre büyüklüğü

İstenilen ağırlıklı kapsama ölçüsü $\alpha(\lambda) = 0, 10$ iken, algoritmamız, DBA, tarafından üretilen temsili kümelerin ortalama ağırlıklı kapsama ölçüsü $\alpha_a(\lambda) = 0, 1056$ olarak hesaplanmıştır. İstenilen kalite ölçüsünden farklı olmasına rağmen hedeften sapma çok azdır. $\alpha_a(\lambda) = 0, 1056$ iken tahmini kategoriler ile hesaplanan ağırlıklı kapsama ölçüsü ise $\alpha_p(\lambda) = 0, 1050$ 'dir. Aradaki farkın az olması gerçek kategoriler ile tahmini kategoriler arasında az farkın olduğunu, bu da tahmin yöntemimiz olan DAA'nin iyi çalıştığını göstermektedir.

Bu incelemelere ek olarak, DBA hipotetik noktaların gerçek kategorilerinin algoritma boyunca bilindiği varsayımı altında tekrar çalıştırılmış ve bu koşullar altında farklı temsili altkümeler üretilmiştir. Karşılaştırma amaçlı yapılan bu çalışmada üretilen temsili altkümelerin ortalama kapsama ölçüsü $\alpha_t(\lambda) = 0, 1031$ olarak hesaplanmıştır. Öyle ki, gerçek kategoriler bilindiği durumda dahi erişilebilen kapsama ölçüsü değeri DBA tarafından ulaşılan kapsama ölçüsü değerine çok yakındır. Burada, yine istenilen ağırlıklı kapsama ölçüsünden sapmanın nedeni ise baskın nokta kümesinin L_p yüzeyi ile yaklaşık olarak tanımlanarak hipotetik noktaların üretilmesinden kaynaklıdır. Tüm baskın nokta kümesinin bulunmasının işlem maliyeti düşünüldüğünde yapılan hata kabul edilebilir seviyede, çok küçüktür. Şekil 30, DBA'nın performansının baskın nokta kümesi arttıkça iyileştiğini göstermektedir. DBA, baskın noktaların ortalama %1,25'ini üreterek baskın nokta kümesini temsil ederken, bu oran baskın nokta sayısı arttıkça %0,33 değerine kadar hızla düşmektedir. Bu da, temsili altkümenin büyüklüğünün, $|R|$, baskın nokta sayısına, N , oranının, N arttıkça düştüğünü göstermektedir.

4.4 TSGA ve TSGA-II Yaklaşımları ile Deneyler

Bölüm 3.3'te geliştirilen TSGA ve TSGA-II, karar verici tarafından istenilen kapsama ölçüsüne sahip altkümeler üretmeyi garantileyen yaklaşımlardır. Bu yaklaşımlar, sadece kapsama ölçüsünü değil aynı zamanda çözümlerin dağılım özelliklerini, baskın nokta kümesinin tipik özelliklerini de dikkate almaktadır. TSGA ve TSGA-II yaklaşımlarının değerlendirilmesi için kapsamlı deneyler yapılmış, sonuçlar bu bölümde sunulmuştur.

ÇATP'ler için adım adım baskın nokta üreten yaklaşımlar, her adımda baskılanan veya daha önceki bulunan baskın noktalar tarafından temsil edilen bölgeleri arama uzayından çıkararak ilerler. Bu nedenle, dışbükey olan arama bölgesinde yeni çözüm aranırken en etkin yollardan biri, amaç fonksiyonu uzayını önce küçük parçalara ayırmak ve daha sonra bu parçaları sırasıyla gezerek istenilen çözümü aramaktır. Ceyhan vd. (2019) tarafından geliştirilen mevcut yaklaşımlar SBA ve TDA, Lokman ve Köksalan (2013) tarafından önerilen altmodel bölme yöntemi ile uygulansa da en son geliştirdiğimiz TSGA ve TSGA-II yaklaşımları daha etkin çalıştığı gözlemlenen Dächert vd. (2017) tarafından önerilen bölme-arama yöntemini kullanmaktadır. Bu sebeple, deneylerimizde adil bir kıyaslama yapılabilmesi için bütün yaklaşımlar Dächert vd. (2017) tarafından önerilen yöntem ile yeniden kodlanmıştır. 2, 3, 4 ve 5 amaçlı sırt çantası ve atama problemleri üzerinde yapılan deneylere ait sonuçlar ekte sunulmaktadır. Test edilen problemler ölçeklendirilmiş olup, $\Delta \in \{0, 05; 0, 10; 0, 15; 0, 20; 0, 25\}$ değerleri ile 5 farklı kapsama ölçüsü kullanılmıştır. Deneyler Intel®Core™i7-4770S CPU@ 3.10 GHz, 16 MB RAM ve Windows 10 işletim sistemli bir bilgisayarda yapılmıştır. Bütün algoritmalar C++ dilinde kodlanmıştır ve bütün matematiksel modeller IBM ILOG Cplex 12.7 kullanılarak çözülmüştür. Deneylerimizde elde edilen altkümelerin büyüklüğünü değerlendirmek amacı ile her bir problemde verilen bir kapsama ölçüsünü, Δ , sağlayabilmek için gerekli en az temsilci sayısı da bulunmuştur. Bu sayı, tüm baskın nokta kümesinin bilindiği problemlerde (P_{Δ}^*) modeli çözülerek bulunmuştur ancak deneylerimizde bu noktaların bilinmediği varsayılmıştır.

(P_{Δ}^*):

$$\min \sum_{j=1}^N v_j \quad (23)$$

$$\text{Kısıtlar} \quad (z_i^l - z_i^j)u_{lj} \leq \Delta \quad \forall i, j, l \quad (24)$$

$$\sum_{l=1}^N u_{lj} \leq Nv_j \quad \forall j \quad (25)$$

$$\sum_{j=1}^N u_{lj} = 1 \quad \forall l \quad (26)$$

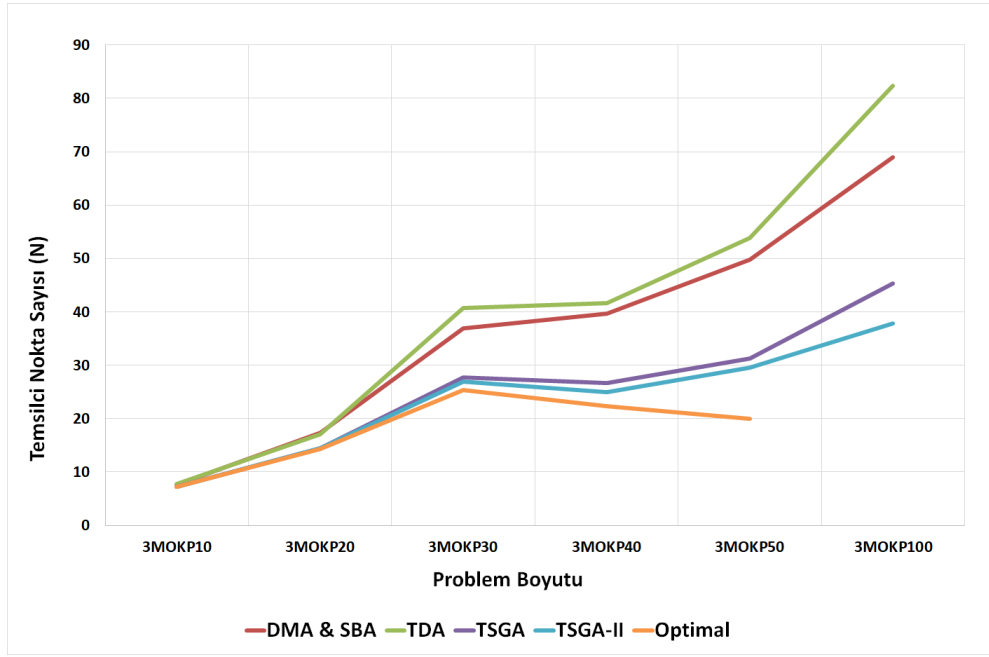
$$v_j, u_{lj} \in \{0, 1\} \quad i = 1, \dots, m \quad j, l = 1, \dots, N \quad (27)$$

Bu modelde, ikili değişken v_j eğer baskın nokta $\mathbf{z}^j = (z_1^j, \dots, z_m^j)$ temsilci olarak seçilmişse

1, aksi takdirde sıfır değerini almaktadır. u_{ij} ikili değişkeni ise, $\mathbf{z}^l = (z_1^l, \dots, z_m^l)$ baskın noktası \mathbf{z}^j tarafından temsil ediliyorsa ancak 1 değerini almaktadır. Bu model çözülerek, N noktadan oluşan tüm baskın nokta kümesinin, \mathbf{Z}^B , d_Δ - temsili olan en küçük alt kümesi bulunmaktadır. Deneylerimizde, bu kümenin büyüklüğü, $\sum_{j=1}^N v_j^*$, optimal temsilci sayısı olarak raporlanmıştır.

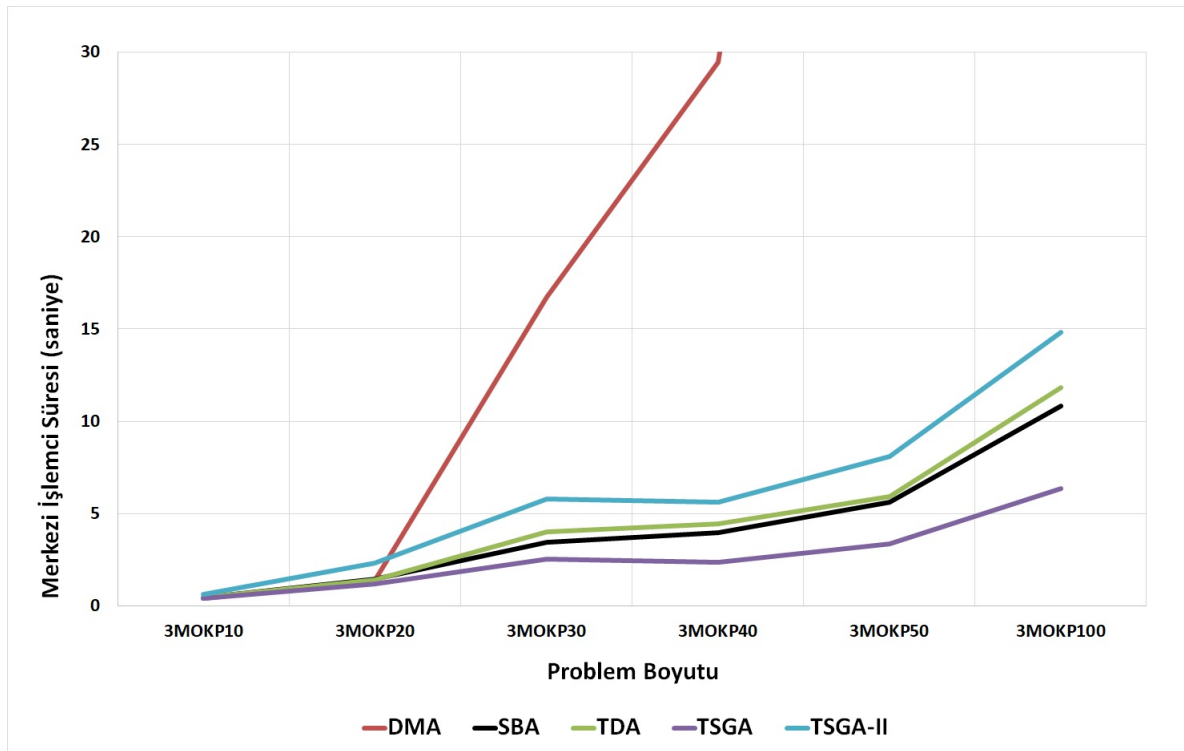
Tablolar 6-11'de her bir problem türünün farklı boyutları için onar adet problemin ortalama değerleri verilmiştir. Bu sonuçları grafiklerle özetlemek gerekirse, ilk olarak, Şekil 31'de 3 amaçlı sırt çantası problemlerinde TSGA ve TSGA-II yaklaşımlarının çözüm sayısı bakımından sergilediği iyileştirmeler yansıtılmıştır. Görüldüğü gibi TSGA ve TSGA-II yaklaşımlarının ürettikleri temsilci nokta sayıları mevcut yaklaşımlardan hep daha az olmaktadır. Ek olarak, problem boyutu arttıkça aradaki fark açılmakta ve iyileşmenin büyük boyutlu problemlerde daha da belirginleştiği görülmektedir. Örneğin, 100 parçalı problemlerde, TSGA tarafından üretilen ortalama temsilci nokta sayısı SBA ve TDA tarafından üretilen nokta sayılarından sırasıyla %34 ve %45 daha az olmuştur. Yine aynı problemlerde, TSGA-II ise TSGA'den ortalama %17 daha az temsilci nokta üreterek izin verilen kapsama hatasını sağlamıştır. Ayrıca, TSGA-II bütün problem boyutlarında ($P_{0,05}^*$) modeli tarafından üretilen optimal çözüm sayısına en yakın sonuçları verirken, 10 parçalı ve 20 parçalı problemlerde optimal temsilci kümesini üretmiştir.

Şekil 32'de ise 3 amaçlı sırt çantası problemlerinde TSGA yaklaşımının çözüm süresi açısından yapmış olduğu iyileştirmeler görülmektedir. Problem boyutu arttıkça, TSGA yaklaşımının verimliliği daha da etkili olmaktadır. Örneğin, 100 parçalı problemlerde, TSGA yaklaşımının ortalama çözüm süresi SBA ve TDA'den sırasıyla %41 ve %46 daha az olmuştur. Son olarak, her adımda çözülen model sayısının fazlalığından dolayı TSGA-II yaklaşımın çözüm süresi mevcut yaklaşımlardan daha fazla olmaktadır. Buna karşılık, TSGA-II'nin çözüm kalitesi açısından kayda değer iyileştirmeler sergilediği Şekil 31'de görülmektedir.



Şekil 31. Farklı boyutlardaki 3 amaçlı sırt çantası problemleri için temsilci nokta sayılarının karşılaştırılması ($\Delta = 0,05$)

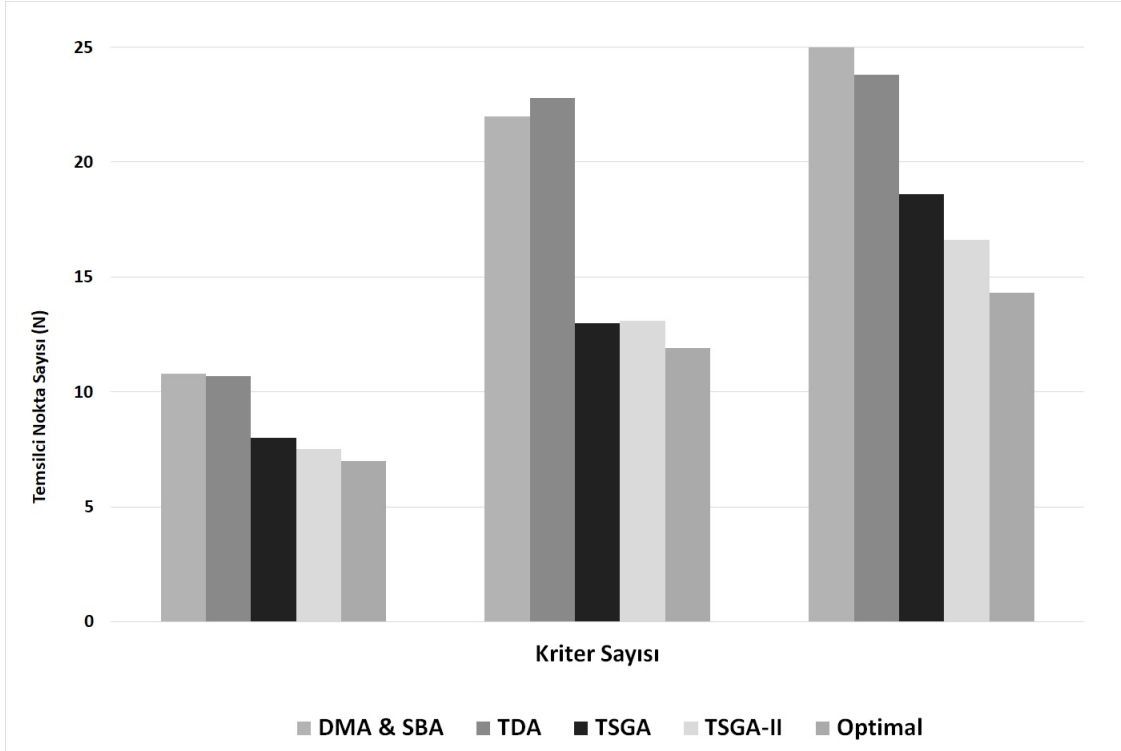
* ÇASP (100 parça) problemlerinde optimal ($P_{0,05}^*$) modeli 4 saatte çözemediği için sonuçlar raporlanamamıştır.



Şekil 32. Farklı boyutlardaki 3 amaçlı sırt çantası problemleri için çözüm sürelerinin karşılaştırılması ($\Delta = 0,05$)

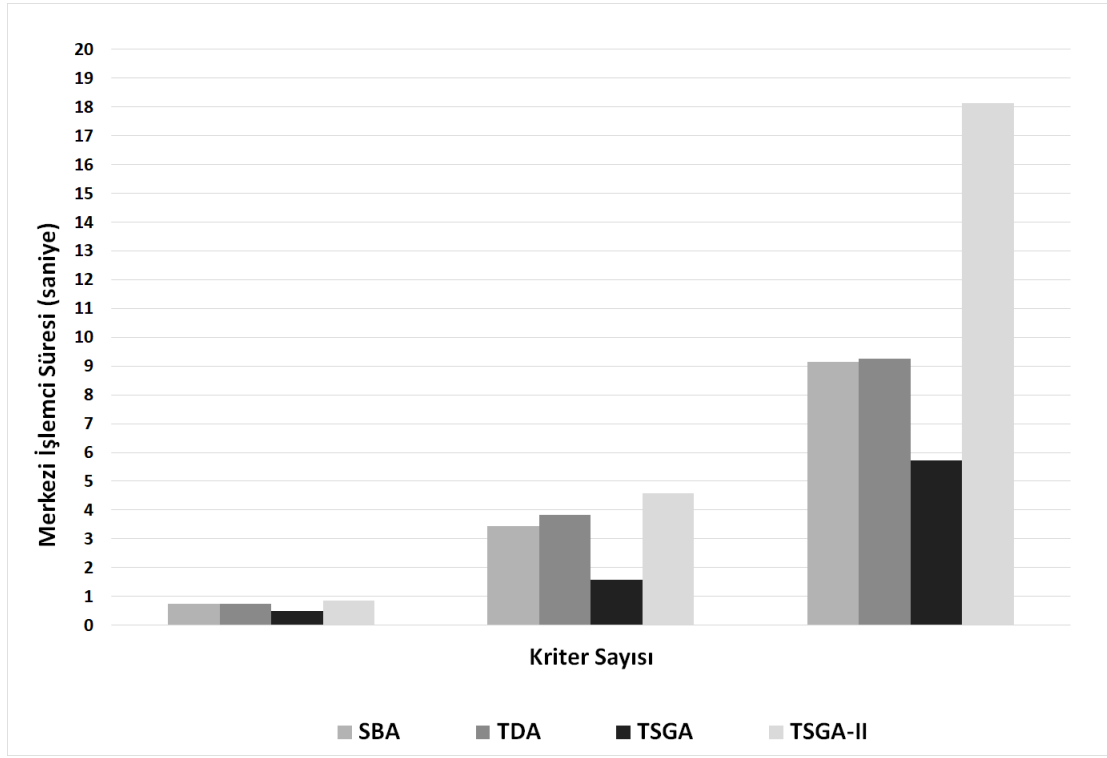
Şekil 33 ve Şekil 34'de ise belirli bir problem boyutunda artan amaç fonksiyonu sayısı için

algoritmaların çözüm sayıları ve süreleri kıyaslanmaktadır. Şekil 33'da, TSGA ve TSGA-II'nin mevcut yaklaşımlardan hep daha az temsilci sayısı ürettiği gözlemlenmektedir. Amaç fonksiyonu sayısı arttıkça, bu iyileşme daha da belirgin hale gelmektedir. Örneğin, 4 amaçlı problemlerde TSGA yaklaşımı, SBA ve TDA'den ortalama %42 daha az temsilci nokta üreterek izin verilen kapsama ölçüsünü sağlamaktadır. Bütün yaklaşımlar arasında optimal temsilci sayısına en yakın çözümleri yine TSGA-II üretmektedir. 3, 4 ve 5 amaçlı problemlerde TSGA-II optimal temsilci sayısından sırasıyla sadece %7, %10 and %16 daha fazla nokta üretmektedir.



Şekil 33. 20 parçalı 3, 4 ve 5 amaçlı sırt çantası problemleri için temsilci nokta sayılarının karşılaştırılması ($\Delta = 0, 10$)

Benzer olarak, Şekil 34'de amaç fonksiyonu sayısı arttıkça TSGA yaklaşımının çözüm süresindeki iyileştirmelerinin de arttığı görülmektedir. 3 amaçlı problemlerde TSGA'in çözüm süresi iyileştirmesi SBA ve TDA'e göre %33 iken, bu değer 4 amaçlı problemlerde %56'ya çıkmaktadır.



Şekil 34. 20 parçalı 3, 4 ve 5 amaçlı sırt çantası problemleri için çözüm sürelerinin karşılaştırılması ($\Delta = 0, 10$)

Tablo 6. TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (3 amaçlı Sirt Çantası Problemleri deneyleri)

| Problem | N | Δ | Temsilci Nokta Sayısı | | | | | Merkezi İşlemci Süresi (saniye) | | | |
|------------------|---------|----------|-----------------------|-------|-------|---------|---------|---------------------------------|-------|------|---------|
| | | | SBA | TDA | TSGA | TSGA-II | Optimal | SBA | TDA | TSGA | TSGA-II |
| ÇASP (10 parça) | 9,30 | 0,05 | 7,60 | 7,80 | 7,20 | 7,20 | 7,20 | 0,43 | 0,44 | 0,42 | 0,64 |
| | | 0,10 | 6,30 | 6,50 | 5,70 | 5,50 | 5,50 | 0,34 | 0,32 | 0,32 | 0,44 |
| | | 0,15 | 5,00 | 5,30 | 4,50 | 4,30 | 4,20 | 0,22 | 0,25 | 0,25 | 0,32 |
| | | 0,20 | 4,20 | 4,30 | 3,50 | 3,40 | 3,40 | 0,17 | 0,19 | 0,18 | 0,21 |
| | | 0,25 | 3,80 | 3,90 | 3,00 | 3,10 | 3,00 | 0,15 | 0,17 | 0,15 | 0,19 |
| ÇASP (20 parça) | 37,40 | 0,05 | 17,40 | 17,10 | 14,40 | 14,40 | 14,30 | 1,44 | 1,42 | 1,20 | 2,30 |
| | | 0,10 | 10,80 | 10,70 | 8,00 | 7,50 | 7,00 | 0,74 | 0,74 | 0,49 | 0,86 |
| | | 0,15 | 8,10 | 7,90 | 5,30 | 4,80 | 4,60 | 0,55 | 0,47 | 0,31 | 0,46 |
| | | 0,20 | 5,90 | 5,70 | 3,60 | 3,40 | 3,20 | 0,36 | 0,32 | 0,20 | 0,29 |
| | | 0,25 | 4,20 | 4,20 | 2,80 | 2,70 | 2,50 | 0,25 | 0,22 | 0,15 | 0,21 |
| ÇASP (30 parça) | 137,90 | 0,05 | 36,90 | 40,70 | 27,70 | 27,00 | 25,30 | 3,45 | 4,01 | 2,54 | 5,77 |
| | | 0,10 | 19,00 | 19,40 | 12,20 | 11,70 | 9,80 | 1,48 | 1,41 | 0,86 | 1,69 |
| | | 0,15 | 12,50 | 11,90 | 6,80 | 6,60 | 5,40 | 0,85 | 0,73 | 0,45 | 0,77 |
| | | 0,20 | 8,80 | 8,60 | 4,90 | 4,00 | 3,50 | 0,61 | 0,51 | 0,28 | 0,38 |
| | | 0,25 | 5,60 | 5,60 | 3,30 | 3,00 | 2,50 | 0,36 | 0,29 | 0,19 | 0,28 |
| ÇASP (40 parça) | 229,20 | 0,05 | 39,70 | 41,70 | 26,70 | 25,00 | 22,30 | 3,97 | 4,43 | 2,37 | 5,61 |
| | | 0,10 | 20,60 | 20,40 | 10,10 | 9,40 | 7,50 | 1,69 | 1,60 | 0,73 | 1,36 |
| | | 0,15 | 12,80 | 12,50 | 5,70 | 5,10 | 4,50 | 0,97 | 0,85 | 0,37 | 0,59 |
| | | 0,20 | 8,60 | 8,30 | 3,90 | 3,70 | 3,00 | 0,61 | 0,52 | 0,23 | 0,35 |
| | | 0,25 | 5,00 | 4,90 | 3,00 | 2,60 | 2,30 | 0,34 | 0,29 | 0,16 | 0,23 |
| ÇASP (50 parça) | 520,90 | 0,05 | 49,80 | 53,80 | 31,30 | 29,50 | 20,00 | 5,61 | 5,92 | 3,37 | 8,11 |
| | | 0,10 | 22,70 | 22,80 | 11,30 | 9,90 | 6,75 | 2,18 | 1,95 | 0,92 | 1,63 |
| | | 0,15 | 13,60 | 13,60 | 6,20 | 5,40 | 3,25 | 1,15 | 0,92 | 0,43 | 0,74 |
| | | 0,20 | 9,40 | 9,60 | 4,10 | 3,40 | 2,50 | 0,72 | 0,65 | 0,29 | 0,33 |
| | | 0,25 | 6,00 | 6,00 | 3,00 | 2,80 | 2,00 | 0,48 | 0,38 | 0,20 | 0,31 |
| ÇASP (100 parça) | 3768,20 | 0,05 | 69,00 | 82,30 | 45,30 | 37,80 | - | 10,83 | 11,82 | 6,37 | 14,84 |
| | | 0,10 | 30,20 | 31,80 | 13,30 | 11,40 | - | 3,64 | 3,38 | 1,35 | 2,35 |
| | | 0,15 | 19,10 | 20,00 | 6,80 | 5,70 | - | 1,99 | 1,76 | 0,55 | 1,00 |
| | | 0,20 | 12,80 | 13,10 | 4,30 | 3,80 | - | 1,23 | 1,04 | 0,35 | 0,50 |
| | | 0,25 | 7,80 | 7,90 | 3,90 | 3,00 | - | 0,73 | 0,59 | 0,37 | 0,40 |

* N baskın nokta sayısını, Δ istenilen kapsama ölçüsü değerini göstermektedir.

* ÇASP (100 parça) problemlerinde optimal (P_{Δ}^*) modeli 4 saatte çözülemediği için sonuçlar raporlanamamıştır.

Tablo 7. TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (4 amaçlı Sirt Çantası Problemleri deneyleri)

| Problem | N | Δ | Temsilci Nokta Sayısı | | | | | Merkezi İşlemci Süresi (saniye) | | | |
|-----------------|---------|----------|-----------------------|--------|-------|---------|---------|---------------------------------|-------|-------|---------|
| | | | SBA | TDA | TSGA | TSGA-II | Optimal | SBA | TDA | TSGA | TSGA-II |
| ÇASP (10 parça) | 11,60 | 0,05 | 8,50 | 8,40 | 7,90 | 7,80 | 7,80 | 0,78 | 0,67 | 0,73 | 1,46 |
| | | 0,10 | 6,70 | 6,50 | 5,70 | 5,50 | 5,50 | 0,52 | 0,46 | 0,52 | 0,85 |
| | | 0,15 | 5,20 | 5,40 | 4,50 | 4,40 | 4,20 | 0,35 | 0,35 | 0,41 | 0,65 |
| | | 0,20 | 3,90 | 4,00 | 3,40 | 3,30 | 3,10 | 0,21 | 0,24 | 0,34 | 0,46 |
| | | 0,25 | 2,90 | 3,00 | 2,90 | 2,80 | 2,40 | 0,14 | 0,16 | 0,30 | 0,38 |
| ÇASP (20 parça) | 136,80 | 0,05 | 43,30 | 43,60 | 32,80 | 33,10 | 31,80 | 8,87 | 10,12 | 6,75 | 30,86 |
| | | 0,10 | 22,00 | 22,80 | 13,00 | 13,10 | 11,90 | 3,43 | 3,82 | 1,59 | 4,58 |
| | | 0,15 | 13,70 | 14,40 | 7,60 | 7,40 | 6,40 | 1,68 | 1,82 | 0,77 | 1,71 |
| | | 0,20 | 8,60 | 8,40 | 5,00 | 4,30 | 4,00 | 0,88 | 0,84 | 0,40 | 0,76 |
| | | 0,25 | 5,70 | 5,30 | 3,30 | 3,20 | 2,80 | 0,51 | 0,44 | 0,24 | 0,43 |
| ÇASP (30 parça) | 397,60 | 0,05 | 76,10 | 82,60 | 54,10 | 52,60 | 48,00 | 18,65 | 25,73 | 13,40 | 56,08 |
| | | 0,10 | 33,70 | 34,80 | 17,60 | 17,50 | 13,60 | 6,78 | 6,77 | 2,53 | 7,88 |
| | | 0,15 | 18,70 | 19,00 | 9,00 | 8,40 | 6,38 | 18,26 | 2,73 | 0,85 | 2,06 |
| | | 0,20 | 10,90 | 11,50 | 5,60 | 5,00 | 3,20 | 34,44 | 1,28 | 0,44 | 0,87 |
| | | 0,25 | 6,60 | 7,00 | 3,80 | 3,10 | 2,40 | 0,77 | 0,67 | 0,28 | 0,41 |
| ÇASP (40 parça) | 1808,60 | 0,05 | 120,20 | 128,50 | 76,30 | 72,00 | 39,50 | 40,98 | 51,55 | 21,20 | 121,68 |
| | | 0,10 | 44,20 | 44,90 | 20,50 | 18,80 | 10,00 | 10,25 | 10,68 | 3,13 | 10,07 |
| | | 0,15 | 23,00 | 22,90 | 10,40 | 8,00 | - | 3,77 | 3,45 | 1,22 | 2,35 |
| | | 0,20 | 13,30 | 13,50 | 6,00 | 5,20 | - | 1,77 | 1,64 | 0,52 | 0,98 |
| | | 0,25 | 7,80 | 7,90 | 3,70 | 3,30 | - | 44,80 | 0,74 | 0,31 | 0,47 |

* N baskın nokta sayısını, Δ istenilen kapsama ölçüsü değerini göstermektedir.

* ÇASP (40 parça) problemlerinde optimal (P_{Δ}^*) modeli 4 saatte çözülemediği için sonuçlar raporlanamamıştır.

Tablo 8. TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (5 amaçlı Sirt Çantası Problemleri deneyleri)

| Problem | N | Δ | Temsilci Nokta Sayısı | | | | | Merkezi İşlemci Süresi (saniye) | | | |
|-----------------|--------|----------|-----------------------|-------|-------|---------|---------|---------------------------------|-------|-------|---------|
| | | | SBA | TDA | TSGA | TSGA-II | Optimal | SBA | TDA | TSGA | TSGA-II |
| ÇASP (10 parça) | 16,20 | 0,05 | 14,30 | 14,10 | 13,70 | 13,70 | 13,70 | 2,92 | 2,49 | 2,42 | 7,50 |
| | | 0,10 | 10,50 | 10,70 | 9,30 | 9,30 | 9,30 | 1,65 | 1,51 | 1,33 | 3,20 |
| | | 0,15 | 8,20 | 8,40 | 7,20 | 7,00 | 6,90 | 1,05 | 0,96 | 0,96 | 1,79 |
| | | 0,20 | 6,80 | 6,60 | 5,70 | 5,30 | 5,20 | 0,76 | 0,68 | 0,61 | 1,06 |
| | | 0,25 | 4,60 | 4,40 | 4,70 | 4,30 | 3,90 | 0,38 | 0,34 | 0,44 | 0,65 |
| ÇASP (20 parça) | 161,20 | 0,05 | 55,20 | 54,40 | 46,60 | 46,50 | 44,40 | 31,81 | 37,56 | 28,68 | 177,36 |
| | | 0,10 | 25,00 | 23,80 | 18,60 | 16,60 | 14,30 | 9,13 | 9,25 | 5,73 | 18,12 |
| | | 0,15 | 14,90 | 12,90 | 9,40 | 8,80 | 7,00 | 3,46 | 2,69 | 1,51 | 3,97 |
| | | 0,20 | 8,70 | 8,00 | 4,90 | 4,60 | 3,80 | 1,48 | 1,24 | 0,65 | 1,28 |
| | | 0,25 | 6,30 | 6,30 | 3,60 | 3,50 | 2,40 | 0,87 | 0,80 | 0,41 | 0,69 |

* N baskın nokta sayısını, Δ istenilen kapsama ölçüsü değerini göstermektedir.

Tablo 9. TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (3 amaçlı Atama Problemleri deneyleri)

| Problem | N | Δ | Temsilci Nokta Sayısı | | | | | Merkezi İşlemci Süresi (saniye) | | | |
|-----------------|---------|----------|-----------------------|-------|-------|---------|---------|---------------------------------|-------|------|---------|
| | | | SBA | TDA | TSGA | TSGA-II | Optimal | SBA | TDA | TSGA | TSGA-II |
| ÇAAP (5 parça) | 18,00 | 0,05 | 14,10 | 14,30 | 13,30 | 13,10 | 13,10 | 0,62 | 0,70 | 0,53 | 1,10 |
| | | 0,10 | 10,60 | 11,10 | 9,40 | 8,90 | 8,80 | 0,39 | 0,48 | 0,36 | 0,63 |
| | | 0,15 | 8,20 | 8,30 | 6,80 | 6,70 | 6,60 | 0,29 | 0,33 | 0,24 | 0,44 |
| | | 0,20 | 5,90 | 5,80 | 4,90 | 4,50 | 4,30 | 0,20 | 0,27 | 0,18 | 0,26 |
| | | 0,25 | 4,40 | 4,50 | 4,00 | 3,50 | 3,10 | 0,15 | 0,20 | 0,15 | 0,19 |
| ÇAAP (10 parça) | 189,40 | 0,05 | 48,20 | 53,90 | 34,60 | 33,40 | 31,40 | 4,72 | 5,35 | 2,74 | 8,41 |
| | | 0,10 | 22,40 | 23,80 | 14,00 | 12,30 | 10,30 | 1,72 | 1,82 | 0,91 | 1,92 |
| | | 0,15 | 14,90 | 14,90 | 7,20 | 7,10 | 5,67 | 1,05 | 0,96 | 0,44 | 0,89 |
| | | 0,20 | 9,20 | 9,10 | 5,10 | 4,10 | 3,56 | 0,54 | 0,50 | 0,28 | 0,40 |
| | | 0,25 | 5,90 | 5,70 | 3,30 | 2,90 | 2,44 | 0,33 | 0,28 | 0,18 | 0,26 |
| ÇAAP (15 parça) | 608,50 | 0,05 | 61,50 | 68,40 | 39,30 | 37,20 | 26 | 8,68 | 9,27 | 4,13 | 14,86 |
| | | 0,10 | 27,00 | 28,40 | 14,20 | 11,80 | - | 3,16 | 3,20 | 1,24 | 2,60 |
| | | 0,15 | 16,10 | 17,10 | 7,30 | 5,90 | - | 1,66 | 1,63 | 0,61 | 0,98 |
| | | 0,20 | 10,20 | 10,60 | 4,90 | 3,90 | - | 1,00 | 0,88 | 0,40 | 0,50 |
| | | 0,25 | 6,00 | 6,10 | 3,00 | 2,80 | - | 0,55 | 0,47 | 0,24 | 0,33 |
| ÇAAP (20 parça) | 1908,50 | 0,05 | 78,80 | 92,40 | 56,70 | 48,80 | - | 16,13 | 17,89 | 8,05 | 29,44 |
| | | 0,10 | 31,60 | 34,90 | 16,90 | 14,00 | - | 5,42 | 5,59 | 1,98 | 4,45 |
| | | 0,15 | 21,50 | 21,40 | 8,60 | 7,00 | - | 3,17 | 2,78 | 0,94 | 1,67 |
| | | 0,20 | 13,50 | 13,60 | 4,80 | 3,90 | - | 1,70 | 1,45 | 0,51 | 0,72 |
| | | 0,25 | 8,50 | 8,50 | 3,70 | 2,90 | - | 0,96 | 0,79 | 0,35 | 0,50 |

* N baskın nokta sayısını, Δ istenilen kapsama ölçüsü değerini göstermektedir.

* ÇAAP (15 ve 20 parça) problemlerinde optimal (P_{Δ}^*) modeli 4 saatte çözülemediği için sonuçlar raporlanamamıştır.

Tablo 10. TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (4 amaçlı Atama Problemleri deneyleri)

| Problem | N | Δ | Temsilci Nokta Sayısı | | | | | Merkezi İşlemci Süresi (saniye) | | | |
|-----------------|---------|----------|-----------------------|--------|--------|---------|---------|---------------------------------|--------|-------|---------|
| | | | SBA | TDA | TSGA | TSGA-II | Optimal | SBA | TDA | TSGA | TSGA-II |
| ÇAAP (5 parça) | 37,50 | 0,05 | 27,10 | 27,30 | 27,00 | 26,70 | 26,50 | 3,86 | 3,50 | 2,70 | 11,03 |
| | | 0,10 | 19,10 | 19,30 | 16,90 | 16,30 | 16,00 | 2,32 | 2,30 | 1,48 | 4,47 |
| | | 0,15 | 13,30 | 13,10 | 10,70 | 9,70 | 9,40 | 1,48 | 1,32 | 0,85 | 1,86 |
| | | 0,20 | 9,40 | 9,60 | 7,80 | 7,10 | 5,70 | 0,88 | 0,83 | 0,54 | 1,09 |
| | | 0,25 | 6,80 | 6,70 | 5,50 | 4,90 | 3,70 | 0,53 | 0,49 | 0,36 | 0,59 |
| ÇAAP (10 parça) | 1527,00 | 0,05 | 145,40 | 162,30 | 104,30 | 100,40 | 75,33 | 54,35 | 65,58 | 33,45 | 257,31 |
| | | 0,10 | 50,90 | 55,20 | 29,50 | 26,50 | - | 13,65 | 14,93 | 5,44 | 22,19 |
| | | 0,15 | 26,80 | 27,60 | 14,50 | 12,20 | - | 5,29 | 5,26 | 1,81 | 5,53 |
| | | 0,20 | 15,30 | 15,90 | 8,00 | 6,70 | - | 2,38 | 2,30 | 0,75 | 1,68 |
| | | 0,25 | 9,30 | 9,10 | 5,10 | 4,10 | - | 1,15 | 0,98 | 0,45 | 0,74 |
| ÇAAP (15 parça) | 6351,80 | 0,05 | 233,40 | 272,90 | 147,30 | 134,00 | - | 151,03 | 176,79 | 52,24 | 668,63 |
| | | 0,10 | 69,60 | 76,40 | 33,50 | 29,40 | - | 30,52 | 33,54 | 7,03 | 43,69 |
| | | 0,15 | 35,00 | 35,90 | 13,00 | 11,20 | - | 11,12 | 11,14 | 1,97 | 6,95 |
| | | 0,20 | 19,70 | 20,60 | 7,50 | 6,20 | - | 4,77 | 4,62 | 0,93 | 2,32 |
| | | 0,25 | 11,00 | 11,70 | 4,90 | 3,80 | - | 2,26 | 1,82 | 0,55 | 0,97 |

* N baskın nokta sayısını, Δ istenilen kapsama ölçüsü değerini göstermektedir.

* ÇAAP (10 ve 15 parça) problemlerinde optimal (P_{Δ}^*) modeli 4 saatte çözemediği için sonuçlar raporlanamamıştır.

Tablo 11. TSGA, TSGA-II, SBA, TDA ve optimal çözümlerin karşılaştırılması (5 amaçlı Atama Problemleri deneyleri)

| Problem | N | Δ | Temsilci Nokta Sayısı | | | | | Merkezi İşlemci Süresi (saniye) | | | |
|-----------------|---------|----------|-----------------------|--------|--------|---------|---------|---------------------------------|--------|--------|---------|
| | | | SBA | TDA | TSGA | TSGA-II | Optimal | SBA | TDA | TSGA | TSGA-II |
| ÇAAP (5 parça) | 50,00 | 0,05 | 36,60 | 36,70 | 35,80 | 35,70 | 35,30 | 10,25 | 10,24 | 8,73 | 43,74 |
| | | 0,10 | 22,10 | 23,30 | 21,00 | 20,30 | 20,00 | 19,94 | 5,87 | 3,68 | 14,12 |
| | | 0,15 | 14,70 | 14,90 | 12,80 | 11,70 | 11,90 | 5,94 | 3,19 | 1,91 | 4,86 |
| | | 0,20 | 10,30 | 11,20 | 8,30 | 8,00 | 7,10 | 1,77 | 2,06 | 0,97 | 2,30 |
| | | 0,25 | 7,60 | 8,00 | 6,30 | 5,60 | 4,90 | 1,01 | 1,12 | 0,68 | 1,14 |
| ÇAAP (10 parça) | 3368,20 | 0,05 | 371,30 | 417,20 | 279,70 | 272,20 | - | 709,61 | 848,09 | 445,72 | 7376,14 |
| | | 0,10 | 99,00 | 108,90 | 60,60 | 56,60 | - | 127,65 | 130,43 | 40,66 | 303,68 |
| | | 0,15 | 43,40 | 46,10 | 23,40 | 20,90 | - | 33,47 | 30,93 | 7,46 | 36,95 |
| | | 0,20 | 24,10 | 25,60 | 12,30 | 10,90 | - | 13,36 | 12,16 | 2,88 | 8,76 |
| | | 0,25 | 14,00 | 15,00 | 8,00 | 7,00 | - | 5,08 | 4,90 | 1,39 | 3,46 |

* N baskın nokta sayısını, Δ istenilen kapsama ölçüsü değerini göstermektedir.

* ÇAAP (10 parça) problemlerinde optimal (P_{Δ}^*) modeli 4 saatte çözemediği için sonuçlar raporlanamamıştır.

5 SONUÇLAR

ÇATP'ler için baskın nokta sayısı, problem büyüklüğü ile hızla arttığı için istenilen özelliklere sahip temsili küme bulmak önemli bir problemdir. İstenilen özellikler, uygulamadan uygulamaya farklılık gösterebilir ancak baskın noktaların dağılım özelliklerini bilmek ya da yaklaşık olarak tahmin edebilmek her koşulda karar verici açısından önem taşır.

Bu çalışmada ilk olarak, ÇATP'ler için tüm baskın nokta kümesini üreten yaklaşımlar geliştirilerek web-tabanlı bir karar destek sistemi tasarlanmıştır. <http://www.onlinemoco.com/MOIP/> adresinden ulaşılabilen bu sistem tüm araştırmacıların erişimine ve kullanımına açıktır. Sadece baskın nokta kümesi değil aynı zamanda istenilen kalite ölçüsüne göre temsili altkümeler de üretilebilmektedir. Geliştirilen bu karar destek sistemi kapsamında, ilk olarak tüm baskın noktaların üretilmesinde kullanılacak yöntem geliştirilmiştir. Daha sonra, bulunan noktaların karar vericiye sunulması aşamasında kullanılacak görselleştirme araçları tasarlanmış, geliştirilen bu sistemin internet-tabanlı bir uygulama olarak sunulması için geliştirilen yazılımlar sunulmaktadır. Bu kapsamda, tüm araştırmacıların erişimine açık farklı problemler için üretilen baskın nokta kümelerini içeren dijital kütüphane de tasarlanmıştır.

Geliştirilen yazılım ve kullanılan görselleştirme araçları, ÇATP'ler için baskın noktaların çözüm uzayında eşit aralıklarla dağılmadığını, bununla beraber baskın nokta kümesinin tipik bazı özelliklere sahip olduğu gözlenmiştir. Bu nedenle, ÇATP'ler için baskın nokta kümesini dağılım özellikleri ile beraber iyi temsil eden altkümeler üreten algoritmalar tasarlanmıştır. Bu kapsamda, ilk olarak ÇATP'ler için baskın nokta kümeleri üretilmiş, tanımlanan yoğunluk ölçüsü ile çözümlerin amaç fonksiyonu uzayına dağılımı incelenmiş ve ortak özellikler tespit edilmiştir. Bu bulgular doğrultusunda, temsili küme bulmak amacı ile ilk geliştirdiğimiz yaklaşım, DBA, baskın nokta kümesi bilinmeyen, ya da üretilmeyen, herhangi bir ÇATP için baskın nokta kümesini yaklaşık olarak temsil eden hipotetik noktalar üretilip bu hipotetik noktaları da yoğunluk seviyelerine göre kategorilere ayırmaktadır. Daha sonra, DBA bu hipotetik noktalar içerisinde yeni önerdiğimiz ağırlıklı kapsama ölçüsüne göre temsilciler seçmektedir. Son olarak, temsilci seçilen hipotetik noktalara yakın gerçek baskın noktalar üretilmiştir. Yapılan uygulamalar, bu yöntem ile bulunan temsili kümelerin kalitesinin istenilen kalite seviyesine çok yakın olduğu göstermiştir.

Pratik çözümler üretebilen ve bu nedenle büyük problemlere uygulanabilecek DBA, yoğunluk açısından bir çok bilgi sunmak ile beraber kalite ölçüsü açısından performans garantisi sunmamaktadır. Bu nedenle, karar vericinin dahil olduğu ve istenilen kalite ölçüsünü garantileyen yeni yöntemler geliştirilmiştir. Bu yöntemler, TSGA (Territory-Excluded Supported Generating Algorithm) ve TSGA-II, DBA gibi dağılım özelliklerini dikkate almakla birlikte farklı bir çözüm yaklaşımı içermektedir. Öyle ki, TSGA baskın nokta kümesinin oluşturduğu yüzeyin tipik özelliklerinden faydalanarak her adımda kullanılacak ve amaç fonksiyonlarının ağırlıklandırılmasıyla elde edilen probleme özgü bir amaç fonksiyonu bulmaktadır. Bu amaç fonksiyonunu her adımda en iyileyen çözümü bulmakta ve daha sonra bu çözüm tarafından temsil edilen veya baskılanan tüm bölgeleri arama uzayından çıkararak ilerlemektedir. TSGA-II, amaç fonksiyonlarından birini seçmekte ve temsili baskın noktaları bu amaç fonksiyonuna göre azalan sırada üretmektedir. Bu sayede, arama uzayının boyutunu azaltmaktadır. Ayrıca, TSGA ve TSGA-II sadece istenilen

kalite ölçüsünü garantilemekle kalmayıp aynı zamanda az baskın nokta üretmek bunu başarmışlardır. Buna ek olarak, TSGA-II iki kriterli problemler için verilen kapsama ölçüsünü sağlayan en küçük kümeyi bulmayı garantilemektedir. Yapılan deneylerde algoritmamızın performansını değerlendirmek amacıyla ile baskın nokta kümelerinin bilinmediği varsayımı altında sonuçlar elde edilmiştir. Daha sonra, bu baskın noktalar sadece performansımızı gösteren kalite ölçüsünün hesaplanmasında kullanılmıştır. Deneylerimizde istenilen kalite ölçüsünü sağlayan az sayıda çözüm bulmak amaçlanırken burada bir üst sınır konmamıştır.

Gelecek çalışmalarda toplam baskın nokta sayısı cinsinden verilen bir bütçe doğrultusunda temsili küme bulan yaklaşımlar kullanılabilir. Bu durumda, bütçe yoğunluk seviyelerine göre ve karar vericinin tercihlerine göre bölgelere paylaştırılabilir.

Baskın noktaların yerlerini ve etrafındaki yoğunluk seviyelerini tahmin etmenin önemli olduğu düşüncesindeyiz. Çünkü bu bilgi sadece karar vericiye sunmak için değil aynı zamanda bir çok algortimanın tasarlanmasında kullanılacak önemli bulgular içermektedir. Bu nedenle, çalışmamızın yeni bir araştırma alanı sunduğuna inanıyoruz.

6 KAYNAKLAR

- Boland, N., Charkhgard, H., Savelsbergh, M. 2016. "The l-shape search method for triobjective integer programming", *Mathematical Programming Computation*, 8 (2), 217-251.
- Boland, N., Charkhgard, H., Savelsbergh, M. 2017. "A new method for optimizing a linear function over the efficient set of a multiobjective integer program", *European Journal of Operational Research*, 260(3), 904-919.
- Ceyhan, G., Köksalan, M., Lokman, B. 2019. "Finding a representative nondominated set for multi-objective mixed integer programs", *European Journal of Operational Research*, 272 (1), 61-77.
- Dächert, K., Klamroth, K. 2015. "A linear bound on the number of scalarizations needed to solve discrete tricriteria optimization problems", *Journal of Global Optimization*, 61 (4), 643-676.
- Dächert, K., Klamroth, K., Lacour, R., Vanderpooten, D. 2017. "Efficient computation of the search region in multi-objective optimization", *European Journal of Operational Research*, 260 (3), 841-855.
- Dhaenens, C., Lemesre, J., Talbi, E. G. 2010. "K-ppm: A new exact method to solve multi-objective combinatorial optimization problems", *European Journal of Operational Research*, 200 (1), 45-53.
- Doğan, I. 2018. "Representing the Nondominated Set with a Small Subset in Multi-objective Mixed Integer Programs", *Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Orta Doğu Teknik Üniversitesi*.
- Duda, R. O., Hart, P. E., Stork, D. G. 1973. *Pattern classification*. New York: Wiley.
- Ehrgott, M, Gandibleux, X. 2000. "A survey and annotated bibliography of multiobjective combinatorial optimization", *OR Spektrum*, 22, 425-460.
- Faulkenberg, S. L., Wiecek, M. M. 2010. "On the quality of discrete representations in multiple objective programming", *Optimization and Engineering*, 11 (3), 423-440.
- Gettinger, J., Kiesling, E., Stummer, C., ve Vetschera, R. 2013. "A comparison of representations for discrete multi-criteria decision problems", *Decision support systems*, 54 (2), 976-985.
- Haimes, Y. 1971. "On a bicriterion formulation of the problems of integrated system identification and system optimization", *IEEE transactions on systems, man, and cybernetics*, 1 (3), 296-297.
- Jorge, J. M. 2009. "An algorithm for optimizing a linear function over an integer efficient set", *European Journal of Operational Research*, 195 (1), 98-103.
- Kirlik, G. Sayın, S. 2014. "A new algorithm for generating all nondominated solutions of multi-objective discrete optimization problems", *European Journal of Operational Research*, 232 (3), 479-488.
- Kirlik, G. Sayın, S. 2015. "Computing the nadir point for multiobjective discrete optimization problems", *Journal of Global Optimization*, 62 (1), 79-99.
- Klamroth, K., Lacour, R., Vanderpooten, D., 2015. "On the representation of the search region in multi-objective optimization", *European Journal of Operational Research*, 245 (3), 767-778.

- Korhonen, P., Wallenius, J. 2008. "Visualization in the multiple objective decision-making framework". Multiobjective optimization. Editör: Branke, J., Deb, K., Miettinen, K. ve Slowinski, R. Berlin: Springer.
- Köksalan, M., Lokman, B. 2009. "Approximating the nondominated frontiers of multi-objective combinatorial optimization problems", *Naval Research Logistics*, 56 (2), 191-198.
- Köksalan, M., Lokman, B. 2015. "Finding nadir points in multi-objective integer programs", *Journal of Global Optimization*, 62 (1), 55-77.
- Köksalan, M. M. 1999. "A heuristic approach to bicriteria scheduling", *Naval Research Logistics*, 46 (7), 777-789.
- Laumanns, M., Thiele, L., Zitzler, E. 2006. "An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method", *European Journal of Operational Research*, 169 (3), 932-942.
- Lokman, B. 2017. "Çok amaçlı tamsayı programlama problemleri için temsili çözüm üreten yaklaşımların ve kalite ölçülerinin incelenmesi", *Endüstri Mühendisliği Dergisi*, 28 (1), 19-39.
- Lokman, B., Köksalan, M. 2013. "Finding all nondominated points of multi-objective integer programs", *Journal of Global Optimization*, 57 (2), 347-365.
- Lokman, B., Köksalan, M. 2014. "Finding highly preferred points for multi-objective integer programs", *IIE Transactions*, 46 (11), 1181-1195.
- Lokman, B., Köksalan, M., Korhonen, P. J., Wallenius, J. 2016. "An interactive algorithm to find the most preferred solution of multi-objective integer programs", *Annals of Operations Research*, 245 (1-2), 67-95.
- Lokman, B., Köksalan, M., Korhonen, P. J., and Wallenius, J. 2018. "An interactive approximation algorithm for multi-objective integer programs", *Computers Operations Research*, 96, 80-90.
- Lotov, A., V., Miettinen, K. 2008. "Visualizing the Pareto Frontier". Multiobjective optimization. Editör: Branke, J., Deb, K., Miettinen, K. ve Slowinski, R. Berlin: Springer.
- Masin, M., Bukchin, Y. 2008. "Diversity Maximization Approach for Multi-objective Optimization", *Operations Research*, 56 (2), 411-424.
- Özarık, S. S. 2017. "Issues in Selecting a Representative Set for Multi-objective Integer Programs", *Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Orta Doğu Teknik Üniversitesi*.
- Özlen, M., Azizoglu, M. 2009. "Multi-objective integer programming: A general approach for generating all non-dominated solutions", *European Journal of Operational Research*, 199 (1), 25-35.
- Özlen, M., Burton, B. A., MacRae, C. A. 2014. "Multi-objective integer programming: An improved recursive algorithm", *Journal of Optimization Theory and Applications*, 160 (2), 470-482.
- Phelps, S. and Köksalan, M. 2003. "An interactive evolutionary metaheuristic for multiobjective combinatorial optimization", *Management Science*, 49 (12), 1726-1738.
- Przybylski, A., Gandibleux, X., Ehrgott, M. 2008. "Two phase algorithms for the bi-objective assignment problem", *European Journal of Operational Research*, 185 (2), 509-533.
- Przybylski, A., Gandibleux, X., Ehrgott, M. 2010. "A recursive algorithm for finding all nondom-

inated extreme points in the outcome set of a multiobjective integer programme", *INFORMS Journal on Computing*, 22 (3), 371–386.

Sayın, S. 2000. "Optimizing over the efficient set using a top-down search of faces. *Operations Research*", 48 (1), 65–72.

Sayın, S. 2000. "Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming", *Mathematical Programming*, 87 (3), 543-560.

Steuer, R. E. 1986. *Multiple Criteria Optimization: Theory, Computation, and Application*, New York: John Wiley Sons, Inc.

Sylva, J., Crema, A. 2004. "A method for finding the set of non-dominated vectors for multiple objective integer linear programs", *European Journal of Operational Research*, 158 (1), 46-55.

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*, 180 (3), 1011-1027.

Zitzler, E., Thiele, L. 1999. "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach", *IEEE transactions on Evolutionary Computation*, 3 (4), 257–271.

TÜBİTAK
PROJE ÖZET BİLGİ FORMU

| | |
|---|--|
| Proje Yürütücüsü: | Dr. Öğr. Üyesi BANU LOKMAN |
| Proje No: | 215M844 |
| Proje Başlığı: | Çok Amaçlı Tamsayı Problemlerinde Baskın Çözümler Üzerine: Analizler, Yaklaşımlar Ve Uygulamalar |
| Proje Türü: | 1001 - Araştırma |
| Proje Süresi: | 30 |
| Araştırmacılar: | MUSTAFA MURAT KÖKSALAN |
| Danışmanlar: | |
| Projenin Yürütüldüğü Kuruluş ve Adresi: | ORTA DOĞU TEKNİK Ü. |
| Projenin Başlangıç ve Bitiş Tarihleri: | 15/04/2016 - 15/10/2018 |
| Onaylanan Bütçe: | 209070.0 |
| Harcanan Bütçe: | 113514.23 |
| Öz: | <p>Günümüzde karmaşık sistemlerde, karar vericiler çoğu zaman birbiri ile çelişen çok amaçlı optimizasyon problemleri ile karşı karşıyadır. Bu problemlerde, genellikle tek bir anlamlı çözüm yoktur. Baskın çözümleri, yani en az bir amaç fonksiyonundan ödün vermeden herhangi bir amaç fonksiyonunda iyileştirme yapılması mümkün olmayan çözümleri bulmak önemlidir. Ancak, baskın çözüm sayısının problem büyüklüğü arttıkça üssel büyümesi nedeniyle; gerçek hayat problemlerinde tüm baskın çözümleri bulmak zor olduğu gibi karar verici açısından da her zaman pratik bir yaklaşım değildir. Bu projede, Çok Amaçlı Tamsayı Problemleri (ÇATP) için farklı yaklaşımlar geliştirilmiş ve web tabanlı bir karar destek sistemi tasarlanmıştır. Bu kapsamda, ilk olarak tüm baskın çözümleri etkin bir şekilde bulan bir algoritma tasarlanmıştır. Daha sonra, yoğunluk analizinde kullanılacak bir ölçü tanımlanmış ve bu yoğunluk ölçüsü kullanılarak ÇATP'ler için ortak dağılım özellikleri elde edilmiştir. Bu özelliklerden faydalanarak, verilen herhangi bir ÇATP için baskın çözümlerin amaç fonksiyonları uzayında olası yerlerini ve dağılımını tahmin eden bir yöntem geliştirilmiştir. Bu bulgular kullanılarak, baskın nokta kümesini iyi temsil eden çözümler üreten algoritmalar geliştirilmiştir. Mevcut algoritmalarından farklı olarak, bu algoritmalar hem baskın nokta kümesinin dağılım özelliklerini hem de karar vericinin tercihlerini yansıtacak altkümeler üretecek şekilde tasarlanmıştır. Farklı ÇATP'ler üzerinde deneyler yapılarak; temsili baskın nokta kümesi üretmek için geliştirilen algoritmaların performansı değerlendirilmiş ve sonuçlarla algoritmaların iyi çalıştığı gösterilmiştir. Proje kapsamında, internet tabanlı ve tüm araştırmacıların erişimine açık araçlar geliştirilmiştir. Bu kapsamda geliştirilen yazılım, karar vericinin belirttiği kalite ölçüleri doğrultusunda çözüm kümeleri üretebilmektedir. Ayrıca karar vericiye çözümlerin sunulmasında kullanılabilecek görselleştirme araçları web-tabanlı uygulama ile beraber tasarlanmış ve farklı ÇATP problemleri için oluşturulan test örneklerinin yayımlanacağı dijital bir test kütüphanesi de oluşturulmuştur.</p> |
| Anahtar Kelimeler: | çok amaçlı tamsayı problemi, baskın nokta, temsili küme, etkileşimli algoritma, görselleştirme aracı |
| Fikri Ürün Bildirim Formu Sunuldu Mu?: | Hayır |

| | |
|----------------------------|---|
| Projeden Yapılan Yayınlar: | <p>1- ÇOK AMAÇLI TAMSAYI PROGRAMLAMA PROBLEMLERİ İÇİN TEMSİLİ ÇÖZÜM ÜRETEN YAKLAŞIMLARIN VE KALİTE ÖLÇÜLERİNİN İNCELENMESİ (Makale - Diğer Hakemli Makale),</p> <p>2- A web-based solution platform for Multi-objective Integer Programs (Bildiri - Uluslararası Bildiri - Sözlü Sunum),</p> <p>3- A Web Application For Solving Multi-objective Integer Programming Problems (Bildiri - Uluslararası Bildiri - Sözlü Sunum),</p> <p>4- Representing the Nondominated Set for Multi-objective Integer Programs (Bildiri - Uluslararası Bildiri - Sözlü Sunum),</p> <p>5- Issues in Selecting a Representative Set for Multi-Objective Integer Programs (Bildiri - Uluslararası Bildiri - Sözlü Sunum),</p> <p>6- ISSUES IN SELECTING A REPRESENTATIVE SET FOR MULTI-OBJECTIVE INTEGER PROGRAMS (Tez (Araştırmacı Yetiştirilmesi) - Yüksek Lisans Tezi),</p> <p>7- ISSUES IN SELECTING A REPRESENTATIVE SET FOR MULTI-OBJECTIVE INTEGER PROGRAMS (Tez (Araştırmacı Yetiştirilmesi) - Yüksek Lisans Tezi),</p> <p>8- Representing the nondominated set with a small subset in multi-objective mixed integer programs (Tez (Araştırmacı Yetiştirilmesi) - Yüksek Lisans Tezi),</p> |
|----------------------------|---|

TÜBİTAK