



# Dinamik Çok Amaçlı Eniyileme Problemleri için Hibrid Çerçevenin İncelenmesi

Berna KİRAZ<sup>1,2,\*</sup>

<sup>1</sup>Fatih Sultan Mehmet Vakıf Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34445, Haliç Yerleşkesi Beyoğlu/İSTANBUL

<sup>2</sup>Marmara Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 34722, Göztepe Kampüsü Kadıköy/İSTANBUL

## Öz

Çok amaçlı evrimsel algoritmalar ve sezgisel seçen üst-sezgiseller ortamda meydana gelebilecek farklı dinamizm tiplerini ele alan adaptif yöntemlerdir. Bu çalışmada, bu yöntemlerin birleştirildiği yapı, dinamik çok amaçlı eniyileme problemlerini çözmek için kullanılmıştır. Bu yapıda üst-sezgiseller popülasyonun bireylerini üretecek olan sezgiselleri seçmek için kullanılır. Sezgisel seçen üst-sezgiseller içinde kullanılan farklı sezgisel seçim yöntemlerinin etkisi ile birlikte önerilen yaklaşımın performansı yapay olarak oluşturulmuş dinamik test problemleri üzerinde deneysel olarak incelenmiştir. Deneysel sonuçlar öğrenme içeren üst-sezgisellerin kullanıldığı yaklaşımın öğrenme içermeyenlere göre daha iyi sonuç verdiğini göstermiştir. Ayrıca, önerilen yaklaşımın literatürde iyi bilinen yöntemlerle karşılaştırıldığında rekabet edebilecek düzeyde sonuçlar verdiği görülmüştür.

## An Investigation of Hybrid Framework for Dynamic Multi-Objective Problems

### Abstract

Multi-objective evolutionary algorithms and selection hyper-heuristics are adaptive methods that can handle different types of dynamism which may occur in the environment. In this study, a hybrid framework combining these methods is presented for solving dynamic multi-objective optimization problems. In this framework, hyper-heuristics are used to select the heuristic that will generate the individuals in the population. The performance of the approach, along with the effect of different heuristic selection methods used in the selection hyper-heuristics, is experimentally examined over a set of dynamic multi-objective optimization problems. The empirical results show that the selection hyper-heuristics with learning perform well in the framework. It is also shown that the proposed approach can compete with the well-known methods from literature.

### Makale Bilgisi

Başyuru: 17/03/2017

Düzeltilme: 03/01/2018

Kabul: 20/01/2018

### Anahtar Kelimeler

Dinamik çok amaçlı eniyileme problemleri  
Üst-sezgiseller  
Çok amaçlı evrimsel algoritmalar

### Keywords

Dynamic multi-objective optimization problems  
Hyper-heuristics  
Multi-objective evolutionary algorithms

## 1. GİRİŞ (INTRODUCTION)

Eniyileme yöntemleri açısından bakıldığında problemin ortamı problemin karar değişkenlerinin tanımlı değerleri, amaç fonksiyonları ve kısıt gibi parçalardan oluşur. Gerçek dünyada birçok önemli önemli mühendislik problemleri ve bilimsel problemler eniyilenmesi gereken birden fazla amaç içermektedir. Genel olarak bu amaçlar birbiriyle çelişki içindedir, öyle ki bir amaçtaki iyileşme diğer amacın kötüleşmesine neden olabilir. Bu nedenle, bu problemlerde tek bir çözüm yoktur ve bu durumda amaç optimal bir çözüm kümesi bulma olmaktadır. Öte yandan, bu problemler zaman içinde değişime uğrayabilirler. Problemlerdeki bu değişimler, problemin parçalarından herhangi birinde ya da birkaçında, tekil ya da eşzamanlı olarak meydana gelebilir. Bu problemler için geliştirilecek eniyileme yöntemleri ortamdaki değişimleri mümkün olduğunca çabuk bir şekilde takip edebilmelidir.

Dinamik çok amaçlı eniyileme problemlerinde amaç fonksiyonları ve/veya kısıtlar zaman içinde değişim göstermektedir. Bu problemler için literatürde birçok meta-sezgisel yöntemler geliştirilmiş [1-3] ve başarılı sonuçlar elde edilmiştir.

\*İletişim yazarı, e-mail: [bkiraz@fsm.edu.tr](mailto:bkiraz@fsm.edu.tr)

Üst-sezgiseller [4] herhangi bir değişiklik gerekmeden bir dizi zor problemin çözümü için tasarlandığından daha genel bir yöntem olarak düşünülmektedir. Bu yöntemler alt seviyedeki sezgiselleri kullanarak arama işlemini gerçekleştirirler [5, 6]. Üst-sezgiseller ile alt seviyedeki sezgiseller arasında bir bariyer olduğu düşünülür ve probleme ait bilgiler bu bariyerden yukarıya geçemezler. Bu özelliği ile üst-sezgiseller sadece alt seviyedeki sezgisellerin değişmesiyle birçok probleme uygulanabilirler. Literatürde tanımlı iki ana tür üst-sezgisel vardır: sezgisel seçen ve sezgisel üreten üst-sezgisel [4]. Bu çalışmada sezgisel seçen üst-sezgiseller kullanılacaktır.

Meta-sezgiseller ile üst-sezgisellerin birlikte kullanıldığı genel bir çerçeve statik çok amaçlı eniyileme problemleri için geliştirilmiş [7] ve başarılı sonuçlar elde edilmiştir. Bu çalışmada evrimsel algoritma ile üst-sezgisellerin birlikte kullanıldığı bu genel çerçeve dinamik çok amaçlı eniyileme problemlerine uygulanmaktadır. Literatürde dinamik çok amaçlı eniyileme problemleri için önerilen yöntemlerin çoğu evrimsel algoritma tabanlıdır. Adaptif yapısı düşünüldüğünde üst-sezgiseller, dinamik ortamlardaki değişimlere, herhangi bir dış müdahale gerektirmeden hızla uyum gösterip, etkin çözümler üretebilirler [8].

Bu çalışmada evrimsel algoritma olarak literatürde iyi bilinen domine edilmemiş sınıflama genetik algoritması (elitist non-dominated sorted genetic algorithm (NSGA-II)) [9] kullanılmıştır. Üst-sezgiseller yeni popülasyonun bireylerini yaratacak olan sezgiseli seçer ve uygular. Deneysel çalışmalarda ilk olarak önerilen yapı için en uygun olanı belirlemek için sezgisel seçme yöntemlerinin başarıma katkısı araştırılmıştır. Bunlara ek olarak, önerilen yapı literatürde önerilen yöntemlerle kıyaslanmıştır. Sonuçlara göre öğrenme içeren sezgisel seçme yöntemleri öğrenme içermeyenlere göre daha başarılı olduğu gözlemlenmiştir. Öte yandan, önerilen yöntem umut verici sonuçlar sergilemiştir.

Bu makalenin geri kalan kısmı şöyle düzenlenmiştir: Bir sonraki bölümde statik ve dinamik çok amaçlı eniyileme problemleri ve üst-sezgisellerle ilgili kısa bir literatür özetiyle birlikte temel kavramlar verilmektedir. 3. Bölüm evrimsel algoritma ile üst-sezgisellerin birlikte kullanıldığı yapıyı tanımlamaktadır. Deneysel çalışmalar ve sonuçlar 4. Bölüm’ de verilmektedir ve bunu makale sonuçlarının ve ileriki çalışma olanaklarının tartışıldığı 5. Bölüm izlemektedir.

## 2. TEMEL KAVRAMLAR (BACKGROUND)

### 2.1. Dinamik Çok Amaçlı Eniyileme Problemleri (Dynamic Multiobjective Optimization Problems)

Çok amaçlı eniyileme problemleri genellikle birbiriyle çelişen iki ya da daha fazla amaç fonksiyonlarından oluşan eniyileme problemleridir. Bu tür problemlerde birden fazla amaç fonksiyonu olduğu için tüm amaç fonksiyonlarını optimize eden tek bir çözümün bulunması mümkün değildir. Bu durumda tek bir optimal çözüm yerine tüm amaç fonksiyonları için kabul edilebilir düzeyde değerler veren bir çözüm kümesinin bulunması hedeflenmektedir [10]. Bu çözüm kümesine Pareto-optimal çözümler kümesi, bu kümedeki her bir elemana Pareto-optimal çözüm denir. Pareto-optimal çözüm (baskın olmayan çözüm) bütün amaç fonksiyonları için kötü olmayan ve en azından biri için diğerlerinden daha iyi olan çözüm olarak tanımlanır. Bu kümedeki hiçbir çözüm diğer çözümler tarafından domine edilmemiştir. Pareto-optimal çözüm kümesinin amaç uzayındaki eşleştirilmesine Pareto cephesi denir.

Dinamik çok amaçlı eniyileme problemlerinde amaç fonksiyonları ya da problemin kısıtları zamana bağlı olarak değişebilir. Dinamik çok amaçlı eniyileme problemleri aşağıdaki gibi tanımlanabilir:

$$\vec{f}(\vec{x}, t) = [f_1(\vec{x}, t), f_2(\vec{x}, t), \dots, f_m(\vec{x}, t)]^T \quad 2.1$$

Burada  $\vec{x}$  karar değişkenlerinin vektörü,  $t$  ise zaman parametresidir.  $\vec{f}: X \rightarrow R^M$  karar değişken uzayından amaç uzayına olan dönüşüm fonksiyonunu gösterir.

Dinamik çok amaçlı eniyileme problemlerindeki kavramlar (Pareto baskınlık, Pareto optimalliği, vb.) statik çok amaçlı eniyileme problemleriyle aynıdır. Dinamik eniyileme problemleri [2] makalesinde dört grupta sınıflandırılmıştır:

**Tip 1:** Optimal karar değişkenleri değişirken, optimal amaç değerleri değişmez.

**Tip 2:** Hem optimal karar değişkenleri hem de optimal amaç değerleri değişir.

**Tip 3:** Optimal karar değişkenleri değişmez, fakat optimal amaç değerleri değişir.

**Tip 4:** Problem değiştiği halde ne optimal karar değişkenleri ne de optimal amaç değerleri değişmez.

Dinamik çok amaçlı eniyileme problemleri için literatürde geliştirilmiş evrimsel algoritmalar şu şekilde sınıflandırılabilir ([11, 12]): çeşitliliği kontrol eden yöntemler, bellek tabanlı yöntemler, çok popülasyonlu yöntemler.

Çeşitliliği kontrol eden yöntemler ortam değiştiğinde çeşitliliği getiren ya da süreç boyunca çeşitliliği koruyan stratejiler içerirler. Ortam değiştiğinde çeşitliliği artıran yöntemlerin başında hiper-mutasyon gelmektedir [13]. Bu yöntemde değişim olduktan sonra belli bir süre boyunca mutasyon oranı artırılarak aramaya devam edilir. [1] makalesinde domine edilmemiş sıralama genetik algoritması (elitist non-dominated sorted genetic algorithm (NSGA-II)) dinamik eniyileme problemlerine uyarlanmıştır. Bu makalede dinamiklik iki farklı yaklaşım kullanılarak ele alınmıştır: İlk yöntemde ortam değiştiğinde popülasyonun belli bir kesimi rastgele üretilen çözümler ile yer değiştirmektedir. İkincisinde ise ortam değiştiğinde popülasyonun belli sayıdaki üyesine mutasyon uygulanır ve bu çözümler popülasyondaki bireylerle yer değiştirir. İki yöntemde de eklenen çözümler popülasyondan rasgele seçilmiş bireylerle yer değiştirir.

Bellek tabanlı yöntemlerde genel olarak daha önceki ortamlarda başarılı olmuş çözümler bellekte tutulur ve süreç boyunca popülasyona katılırlar ([14-16]). Dinamik çok amaçlı eniyileme problemleri için [17] makalesinde bellek tabanlı yöntem geliştirilmiştir. Bu yöntemde ortam değiştiğinde yeni ortam için popülasyon sadece bellekten değil, çeşitliliği de muhafaza etmek için büyük miktarda rasgele oluşturulmuş çözümlerden oluşmaktadır.

Dinamik eniyileme problemleri için çok popülasyonlu yöntemler iyi sonuçlar vermektedir [18]. Popülasyonun alt popülasyonlara bölüdüğü bu yaklaşımlar arama uzayının farklı bölgelerinde eş zamanlı olarak arama yaparlar. Dinamik çok amaçlı eniyileme problemleri için geliştirilen çok popülasyonlu evrimsel algoritmalar [3, 19] her bir alt popülasyon bir amaç fonksiyonunun eniyilenmesinden sorumludur. [3] tezinde yapılan çalışmada birden fazla sürüye sahip parçacık sürü optimizasyon algoritması geliştirilmiştir. [19] makalesinde statik ve dinamik çok amaçlı eniyileme problemlerini çözmek için çok popülasyonlu eş evrimsel algoritmaların rekabetçi ve kooperatif tekniklerinin birleştirildiği rekabetçi-kooperatif eş evrimsel algoritma önerilmiştir. [20] makalesinde bellek ve arama popülasyonu adı verilen iki popülasyonlu yaklaşım ile belleğin birlikte kullanıldığı hibrid bir algoritma geliştirilmiştir.

Dinamik çok amaçlı eniyileme için geliştirilmiş algoritmalar ortamda meydana gelen değişimi belirlemek için farklı yöntemler kullanılmaktadır. Bunlardan en basit ve yaygın olanı ise şöyledir: Her adımda daha önceden belirlenmiş sayıda bireyin uygunluk değerleri tekrardan hesaplanır. Herhangi bir amaç fonksiyonunda ya da kısıtta değişiklik varsa ortamın değiştiği kabul edilir [1].

Dinamik çok amaçlı eniyilemenin önemli zorlukları ve gelecekteki yönlerinin detaylı olarak irdelendiği çalışmada [21], önemli zorluğun bir algoritmanın performansını değerlendirirken sonuçları analiz etmek için kıyaslama işlevlerinin, performans ölçümlerinin ve yaklaşımının standartlaştırmak olduğu vurgulanmıştır.

## 2.2. Üst-sezgiseller (Hyper-heuristics)

Tek noktalı arama çerçevesine dayanan yinelemeli sezgisel seçen üst-sezgiseller genel olarak sezgisel seçim ve hareket kabul etme bileşenlerini içerir [5]. Sezgisel seçen üst-sezgiseller yüksek düzeyde çalışır ve önceden tanımlanmış alt seviyedeki sezgiselleri kontrol eder. Her adımda, bir sezgisel seçim yöntemine göre seçilir ve mevcut bireye uygulanarak yeni bir çözüm oluşturulur. Sezgisel seçim yöntemi karar verme sırasında problem alanıyla ilgili özel bilgileri kullanmaz. Yeni üretilen çözüm kabul kriterine göre ya kabul edilir ya da edilmez. Bu süreç sonlanma kriter(ler)i sağlanıncaya kadar devam eder ve süreç sonunda bulunan en iyi çözüm döndürülür.

Literatürde birçok sezgisel seçme yöntemleri önerilmiştir. Bunlardan bazıları şunlardır: basit rastsal (Simple Random - SR), rastsal iniş (Random Descent - RD), rastsal permütasyon (Random Permutation - RP), rastsal permütasyon iniş (Random Permutation Descent - RPD), açgözlü seçim (Greedy Selection - GR), seçim fonksiyonu (Choice Function - CF) [6] ve pekiştirmeli öğrenme (Reinforcement Learning - RL) [22]. Basit rastsal her adımda sezgisel kümesinden bir sezgiseli rastgele geçer ve bir kez uygular. Rastsal iniş rastgele seçilmiş bir sezgiseli çözümü iyileştirdiği sürece uygular. Rastsal permütasyon ilk adımda tüm sezgisellerin bir permütasyonunu oluşturur ve her birini sırasıyla uygular. Rastsal permütasyon iniş sezgiseli rastsal permütasyon gibi seçer fakat seçilen sezgiseli iyileşme olduğu sürece uygular. Açgözlü seçim bir adımda tüm sezgiselleri mevcut çözüme uygular ve en iyi iyileşmeyi sağlayan sezgiseli seçer. Seçim fonksiyonu her bir sezgisel için bir skor değeri tutar ve bu skor değeri üç ölçüme dayanır: sezgisellerin bireysel performansları, sezgiseller arası ikili performansları, sezgiselin en son kullanımdan itibaren geçen süre. Her adımda en yüksek skor değerine sahip sezgisel seçilir. Sezgisel uygulandıktan sonra ise o sezgiselin skoru güncellenir. Pekiştirmeli öğrenme seçildikten ve uygulandıktan sonra iyileştirici / kötüleşen bir çözüm getiren sezgiselin performansını belirlemek için ödül / ceza kavramını kullanır. Bu yöntemde her sezgiselin bir skor değeri vardır ve ilk başta hepsi aynı skor değerine sahiptir. Seçilen sezgisel çözümü iyileştirdiyse skor değeri artırılır, aksi halde azaltılır. Her bir adımda en yüksek skor değerine sahip sezgisel mevcut çözüme uygulanmak üzere seçilir.

Üst-sezgisellerin dinamik tek amaçlı eniyileme problemleri için iyi performans sergilediğini gösteren çalışmalar literatürde mevcuttur [8, 23-25]. Özcan vd. [23] dinamik problemlerine açgözlü seçim üst-sezgiselini uygulamışlardır. [8] makalesinde, sezgisel seçen üst-sezgiseller hareket eden tepeler (Moving peaks benchmark - MPB) kullanılarak oluşturulmuş reel değerli eniyileme problemlerine uygulanmıştır. Bu çalışma öğrenme içeren üst-sezgisellerin sadece dinamik ortamlar için değil aynı zamanda sürekli eniyileme problemleri için de uygulanabilir olduğunu göstermiştir. Uludağ vd. [24] dinamik problemler için dağılım tahmini algoritması (Estimation of Distribution Algorithms (EDA)) ile üst-sezgisellerin birlikte kullanıldığı, çevrimiçi ve çevrimdışı öğrenmeyi birleştiren iki fazlı bir yöntem geliştirmişlerdir. Elde edilen sonuçlara göre, genel olarak iyi sonuç verdiği gözlemlenmiştir. [25] makalesinde ise üst-sezgisellerin gömüldüğü popülasyon tabanlı bir yaklaşım önerilmiştir. Önerilen yaklaşımın davranışı hem yapay olarak oluşturulmuş test problemlerinde hem de gerçek dünya problemlerine uygulanmış ve başarılı sonuçlar elde edilmiştir.

### 2.3. Üst-sezgisel Tabanlı Çok Amaçlı Evrimsel Algoritmalar (Hyper-heuristic Based Multiobjective Evolutionary Algorithms)

Statik çok amaçlı eniyileme problemleri için literatürde az sayıda üst-sezgisel tabanlı yöntemler vardır. Tabu arama [26] bu alanda yapılan ilk çalışmalardan biridir. Bu yöntemde, her bir sezgiselin performansı tek amaca göre değil bireysel amaca göre değerlendirilir. Her adımda popülasyondaki her bir çözüm için bir tane bireysel amaç seçilir. Bu bireysel amaca göre ve sezgisellerin skor değerine göre sezgisel seçilir ve seçilen sezgisel o anki mevcut çözüme uygulanır.

McClymont and Keedwell [27] evrimsel strateji yöntemi içine gömülen Markov zinciri üst-sezgiselini önermişlerdir. Bu yöntem, sezgiseller arasındaki işlem olasılıklarını gösteren Markov zincirini kullanarak her karar adımında en uygun sezgiseli seçer. Buna ek olarak, sezgisel çiftleri arasındaki ağırlıklar pekiştirmeli öğrenme yöntemine göre değiştirilir. Bir sonraki çalışmalarında [7] Markov zincirini meta-sezgisel yöntemlere entegre ederek bu yöntemi daha genel hale getirmişlerdir. Meta-sezgisel olarak, iyi bilinen iki tane çok amaçlı evrimsel algoritma, NSGA-II ve geliştirilmiş güç pareto evrim algoritması (SPEA2) [28], düşünülmüştür. Önerilen yöntem su dağıtım şebeke tasarımı problemi üzerine uygulanarak başarımları gösterilmiştir.

Gomez ve Terashima-Marín [29] NSGA-II tabanlı yeni bir üst-sezgisel geliştirmişlerdir. Bu algoritma bir öğrenme süreci boyunca koşul-eylem kurallarının kombinasyonlarını geliştirir ve sonunda bir dizi Pareto-optimal çok amaçlı üst-sezgiseller üretir.

Kaynak [30]'da, çok amaçlı bir yazılım modülü kümeleme problemini çözmek için hızlı, çok amaçlı üst-sezgisel genetik algoritma önerilmiştir. Yöntem her adımda pekiştirmeli öğrenmeye göre bir sezgisel seçer. Seçilen sezgiseller yeni popülasyonu oluşturmak için uygulanır. Sezgiseller seçim / çaprazlama /

mutasyon kombinasyonları ile tanımlanmıştır. Bu yöntemde farklı kombinasyonlara sahip on iki adet sezgisel kullanılmıştır.

### 3. ÜST-SEZGİSEL TABANLI EVRİMSEL ALGORİTMA

Üst-sezgisellerin evrimsel algoritmalarla (EA) birlikte kullanıldığı genel hibrid yapı [7] çalışmasında önerilmiş ve statik çok amaçlı eniyileme problemine uygulanmıştır. Önerilen yapıda üst-sezgiseller evrimsel algoritma gibi var olan meta-sezgisellere entegre edilmiştir. Bu yapıda üst-sezgiseller sezgisel kümesi içinden uygun olan bir sezgiseli seçer, uygular ve varsa parametre ayarlamalarını yapar.

Evrimsel algoritmalar dinamik çok amaçlı eniyileme problemlerine başarılı şekilde uygulanmıştır [1, 20]. Bunlara ek olarak üst-sezgiseller dinamik eniyileme problemlerindeki değişimlere hızla uyum göstermiş ve etkili çözümler üretebilmişlerdir [8]. Bu nedenlerle bu çalışmada üst-sezgisellerin evrimsel algoritmalarla (EA) birlikte kullanıldığı hibrid yapı dinamik çok amaçlı eniyileme problemlerine uygulanmaktadır.

Bu çalışmada evrimsel algoritma olarak literatürde iyi bilinen domine edilmemiş sınıflama genetik algoritması (elitist non-dominated sorted genetic algorithm (NSGA-II)) [9] kullanılmıştır. Bu çalışmadan kullanılan sezgisel seçme yöntemleri ise şunlardır: basit rastsal (SR), rastsal iniş (RD), rastsal permütasyon (RP), rastsal permütasyon iniş (RPD), pekiştirmeli öğrenme (RL) ve Markov zinciri üst-sezgiseli (MCHH). Bu yapının sözde kodu Algoritma 1'de verilmiştir.

**Algoritma 1** NSGA-II ile üst-sezgisellerin birlikte kullanıldığı yapının sözde kodu

1.  $t = 0$
2. Başlangıç çözüm adaylarını üret  $P_0$
3.  $P_0$  daki bireylerin uygunluk değerlerini hesapla
4.  $P_0$  popülasyonunu domine edilmemiş sıralama prosedürünü kullanarak farklı domine edilmemiş sınıflara ayır
5.  $P_0$  popülasyonundaki her bir birey için belirli bir çözümü çevreleyen çözümlerin yoğunluğunu ölçen kalabalık mesafeyi (crowding distance) hesapla
6. **WHILE** (sonlanma ölçütüyle karşılaşmadığı sürece) **DO**
7. Kalabalık turnuva seçimi yöntemini kullanarak ebeveyn seç  $M_t = \text{seç}(P_t)$
8.  $h$  sezgiselini ( $H_1, \dots, H_n$ ) sezgisel setinden seç
9.  $h$  sezgiselini kullanarak yeni çözüm adayları üret  $Q_t = h(M_t)$
10.  $Q_t$  daki bireylerin uygunluk değerlerini hesapla
11. İki popülasyonu birleştirerek yeni bir popülasyon ( $R_t = P_t \cup Q_t$ ) oluştur.
12.  $R_t$  popülasyonunu domine edilmemiş sıralama prosedürünü kullanarak farklı domine edilmemiş sınıflara ayır.
13. Yeni popülasyonun ( $P_{t+1}$ ) bireylerini  $R_t$  daki ilk cepheden başlayarak seç. Eklenen son cephede yeni popülasyonun kalan kısmından fazla ise kalabalık uzaklık atama yöntemini kullanarak yeni popülasyonda yer alacak bireyleri seç.
14.  $t = t + 1$
15. **ENDWHILE**

Dinamik eniyileme problemleri için var olan yapıya ek adımlar ilave edilmiştir. İlk olarak ortamda değişiklik olup olmadığını belirlemek için bir algılama mekanizması kullanılmıştır. Bu mekanizmada, önceden belirlenmiş sayıda rasgele bireyler seçilir ve tekrardan değerlendirilir. Herhangi bir amaç fonksiyonunda ya da kısıtlarda bir değişiklik varsa, ortamda değişiklik olduğu belirtilir. Değişimden

sonra, tüm popülasyonun yeniden değerlendirilir. Daha sonra popülasyon domine edilmemiş sıralamaya göre sıralanır ve kalabalık mesafeleri yeniden hesaplanır.

İlerleyen bölümde bu yapı içinde kullanılan sezgiseller ve öğrenme içeren sezgisel seçme yöntemleri için sezgisellerin performanslarını ölçen metrikler anlatılmıştır.

### 3.1. Sezgiseller (Heuristics)

Bu çalışmada, sezgiseller olarak genetik algoritma (GA) ve diferansiyel evrim algoritmasında (DE) yer alan operatörler kullanılmıştır. Tanımlanan sezgisellerin genel yapısı seçim, çaprazlama ve mutasyondur. Çok amaçlı genetik algoritmaların iyi bir yakınsama hızı sağladığı belirtilirken [31], çok amaçlı diferansiyel evrim algoritmaları uzun bir süreçten sonra iyi dağılmış bir Pareto cephesi üretebilir. Bu nedenlerle, bu yapıda hem genetik hem diferansiyel evrim algoritmalarında yer alan operatörler kullanılmaktadır.

GA operatörleri için seçim, çaprazlama ve mutasyon olarak sırasıyla kalabalık turnuva seçimi, simüle edilmiş ikili çaprazlama ve polinomsal mutasyon kullanılmaktadır. Bu grupta farklı mutasyon olasılık değerleri ve dağılım indekslerine sahip beş farklı sezgisel tanımlanmıştır. Bu sezgisellerdeki çaprazlama olasılık değerleri aynıdır. Sezgisellerdeki parametre değerleri Tablo 3-1'de verilmiştir.

**Tablo 3.1.** Sezgisellerin parametre atamaları

Sezgiseller	$P_c$	$\eta_c$	$P_m$	$\eta_m$
$h_1$	0.9	15	1/n	20
$h_2$	0.9	15	2/n	20
$h_3$	0.9	15	2/n	4
$h_4$	0.9	15	4/n	20
$h_5$	0.9	15	4/n	4

DE için üç farklı operatör kullanılmıştır: *DE/rand/1/bin*, *DE/target-to-best/1/bin* ve *DE/rand/1/Either-Or* [32]. *DE/rand/1/bin* de, mutasyona uğramış vektör aşağıdaki gibi oluşturulur:

$$\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \quad 3.1$$

Burada,  $\vec{X}_{r_1^i,G}$ ,  $\vec{X}_{r_2^i,G}$ , ve  $\vec{X}_{r_3^i,G}$  o anki popülasyondan rasgele seçilmiş 3 farklı vektörü gösterir.  $F$  ise fark vektörlerini ölçeklendirmek için kullanılan ve ölçeklendirme parametresi olarak adlandırılan pozitif bir parametredir. Mutasyon işleminden sonra hedef vektörden ( $\vec{X}_{i,G}$ ) ve ona karşılık gelen mutant vektörden ( $\vec{V}_{i,G}$ ), ikiterimli çaprazlama yöntemi ile yeni vektör ( $\vec{U}_{i,G}$ ) oluşturulur.

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if } (rand_j[0,1] \leq CR) \text{ or } (j = j_{rand}), j = 1, 2, \dots, D \\ x_{i,G}^j & \text{otherwise} \end{cases} \quad 3.2$$

Burada CR çaprazlama olasılık değeridir ve  $[0,1]$  aralığında olan sabit bir değerdir.  $j_{rand}$  ise  $[1, D]$  ( $D$  karar vektörlerinin boyutudur.) aralığında seçilen rasgele bir tam sayıdır.

*DE/target-to-best/1/bin* ise mutasyon işlemi aşağıdaki gibidir:

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F \cdot (\vec{X}_{best,G} - \vec{X}_{i,G}) + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \quad 3.3$$

Burada,  $\vec{X}_{best,G}$  mevcut popülasyondaki en iyi çözümü ifade eder. Çok amaçlı eniyileme problemleri düşünüldüğünde tek bir en iyi çözüm söz konusu değildir. Bu nedenle, bu çalışma en iyi çözüm ilk

seviyedeki cepheden rasgele seçilmiştir. Yine bu yöntemde mutasyondan sonra ikiterimli çaprazlama yöntemi (bkz. Denklem 3.2) kullanılmıştır.

Son olarak *DE/rand/1/Either-Or* yönteminde yeni vektör ya  $p_f$  olasılıkla diferansiyel mutasyon ile oluşturulur ya da  $1 - p_f$  olasılıkla iç noktali çaprazlama ile oluşturulur:

$$u_{i,G}^j = \begin{cases} \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) & , \text{if } (rand_j[0,1] < p_f) \\ \vec{X}_{r_1^i,G} + 0,5 \cdot (F + 1) \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G} - 2 \cdot \vec{X}_{r_1^i,G}) & , \text{otherwise} \end{cases} \quad 3.4$$

Bu çalışmada DE yöntemleri için yaygın olarak kullanılan  $F = 1$  ve  $CR = 0.8$  değerleri göz önüne alınmıştır.

### 3.2. Sezgisel Performans Metrikleri (Heuristic Performance Metrics)

Üst-sezgiseller genellikle tek noktali arama yaparlar ve bir sezgiselin başarısı ürettiği çözümün uygunluk değeri ile belirlenir. Bu çalışmada popülasyon tabanlı çok amaçlı evrimsel algoritma kullanıldığından bir sezgiselin başarısını ölçmek için ürettiği çözümlerin uygunluk değerleri yerine baskınlık içeren metriklerin kullanılması daha uygun olacaktır.

Bu çalışmada, her bir sezgisellerin performanslarını ölçmek için iki farklı metrik düşünülmüştür: baskın olmayan çözümlerin oranı [33] (the ratio of non-dominated individuals (*RNI*)) ve pareto baskınlık (the pareto dominance) [7]. Bu değerler yeni popülasyonun bireylerinin oluşturulmasından hemen sonra hesaplanır. *RNI* metriği yeni popülasyonda yer alan baskın olmayan çözümlerin oranı verir:

$$RNI = \frac{\text{baskın olmayan bireylerin sayısı}}{\text{toplumdaki birey sayısı}} \quad 3.5$$

Pareto baskınlık ilişkisi sezgiseller tarafından üretilen baskın çözümlerin oranını hesaplar. Pareto baskınlık  $h$  sezgiseli kullanılarak üretilmiş olan her yeni birey tarafından domine edilen ebeveyn bireylerin ortalama oranı olarak hesaplanır:

$$p(h, \mu, \lambda) = \frac{\sum_{\forall a \in \lambda \forall b \in \mu} \text{dom}(a,b)}{|\mu| \times |\lambda|} \quad 3.6$$

Burada  $\text{dom}(a,b) = \begin{cases} 1, a < b \\ 0, \text{otherwise} \end{cases}$ ,  $a$  ve  $b$  sırasıyla yeni üretilen çocuk popülasyondaki bireyi ve ebeveyn popülasyondaki bireyi,  $\lambda$  ve  $\mu$  ise sırasıyla yeni oluşturulan çocuk popülasyonu ve ebeveyn popülasyonu temsil eder.

## 4. DENEYSSEL ÇALIŞMALAR (EXPERIMENTAL STUDY)

Bu çalışmada, deneysel çalışma üç gruptan oluşmaktadır. İlk deney grubunda sezgisellerin performans metriklerinin performans üzerindeki etkileri araştırılıp, üst-sezgiseller için hangi metriğin en uygun olduğu belirlenmiştir. İkinci grupta ise farklı sezgisel seçme yöntemlerinin performans değerlendirmeleri yapılmıştır. Son grupta literatürden seçilmiş iyi bilinen algoritmalarla önerilen yöntemin karşılaştırma sonuçları verilmiştir.

### 4.1. Deneylerin Organizasyonu (Experimental Design)

Bu bölümde hibrid yapı içinde kullanılan sezgiseller, sezgisellerin performans metrikleri, deneylerde kullanılan dinamik çok amaçlı eniyileme problemleri anlatılmakta ve tüm deneyler için genel parametre ayarlamalarını sunmaktadır.

#### 4.1.1. Parametre Ayarları (Parameter Settings)

Tüm deneylerde kullanılan parametre ayarlamaları şunlardır: popülasyon boyu 100 olarak atanmıştır. Değişimin sıklığı ( $\tau_t$ ) her deney seti için 50 jenerasyondur. Algoritmaların her bir koşulunda, ilk ortamdan sonra 10 adet değişim oluşmuştur. Değişimin şiddeti ( $n_t$ ) ise yüksek seviyeli şiddet için 1, orta seviyede şiddet için 10, düşük seviyeli şiddet için 20 seçilmiştir. Her adımda ortamda değişimi belirlemek için beş adet bireyin uygunluk değeri tekrardan hesaplanır. Her bir test problemi için her algoritma 30 kere koşturulmuştur. Aksi belirtilmediği sürece bu çalışmada kullanılan parametrelere atanan değerler yapılan ön testlerle belirlenmiştir.

Bu çalışmada kullanılan sezgisel seçme yöntemleri şunlardır: basit rastsal (SR), rastsal iniş (RD), rastsal permütasyon (RP), rastsal permütasyon iniş (RPD), pekiştirmeli öğrenme (RL) ve Markov zinciri üst-sezgiseli (MCHH). Pekiştirmeli öğrenme yönteminde, [34] makalesinde önerilen değerler kullanılmıştır: her sezgiselin ilk skor değeri 15 olarak belirlenmiş ve skor değerleri 0 ve 30 aralığında olmalıdır. Sezgiseller uygulandıktan hemen sonra seçilmiş sezgiselin skor değeri şöyle güncellenir: eğer seçilen sezgisel bir iyileşme sağlamışsa, skor değeri bir artırılır, aksi halde 1 azaltılır. Markov zinciri üst-sezgiseli yönteminde, her bir sezgisel çifti arasındaki ağırlıkların ilk değeri  $1/k$  ( $k$  alt seviyede kullanılan sezgisel sayısı) olarak atanır. Pekiştirmeli öğrenmede olduğu gibi, sol seçilmiş sezgiselden şuan seçilen sezgiselle olan ağırlık performansına göre  $\pm 0.01$  olarak güncellenir.

Önerilen yapı literatürde iyi bilinen dinamik NSGAI-A (DNSGAI-A) ve NSGAI-B (DNSGAI-B) metotlarıyla karşılaştırılmıştır. Bu iki algoritma için popülasyon boyutu 100 olarak alınmıştır. Çaprazlama olasılık değeri 0.9, mutasyon olasılık değeri  $1/n$  ( $n$  karar değişken sayısı), çaprazlama ve mutasyon için dağılım indeks değerleri sırasıyla 10 ve 20 olarak alınmıştır. Ortamda bir değişim olduğunda, popülasyon üyelerinin %20'si DNSGAI-A için rasgele üretilen bireylerle, DNSGAI-B için ise mutasyona uğramış bireylerle değiştirilir. DNSGAI-B yönteminde ortam değiştiğinde uygulanan mutasyonda mutasyon olasılık değeri iki katına çıkarılmış, dağılım indeks değeri ise 10 azaltılmıştır. Bu parametrelere [1] makalesinde önerildiği gibi değerler atanmıştır.

#### 4.1.2. Test Problemleri (Benchmark Problems)

Geliştirilen metotların başarımlarını değerlendirmek için farklı özelliklere sahip yedi adet dinamik çok amaçlı eniyileme problemi kullanılmaktadır. Bu problemlerin hiçbirinde kısıt yoktur ve hepsinde 2 adet amaç fonksiyonu vardır. Tablo 4.1' de deneylerde kullanılan dinamik test problemlerinin karar değişkenleri, tipleri ve tanımlamaları verilmiştir. Tüm problemler en küçükleme problemleridir:

$$\text{en küçükle } F(x, t) = [f_1(x, t), g(x, t), h(x, f_1(x, t), g(x, t), t)]^T$$

Bu problemlerin dinamikliği aşağıda tanımlanmış denklem tarafından belirlenir:

$$t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor$$

Burada  $n_t$  ve  $\tau_t$  sırasıyla değişimin şiddetini ve sıklığını gösterir.  $\tau$  ise o anki jenerasyon sayısını belirtmektedir.

#### 4.1.3. Performans Değerlendirme Kriterleri (Performance Evaluation Criteria)

Algoritmaların başarımlarını karşılaştırmak için, sonuçlar doğruluk oranı [3] (accuracy rate -  $acc$ ) ve tersinir kuşak mesafesi [35] (Inverted Generational Distance- $IGD$ ) metrikleri üzerinden verilmiştir. Doğruluk oranı bir eniyileme yönteminin koşturulmasıyla elde edilen Pareto cephesinin hiperhaciminin (hypervolume -  $HV(POF^*(t))$ ) gerçek Pareto cephesinin hiperhacimine ( $HV(POF'(t))$ ) olan oranı olarak hesaplanır:

$$acc_t = \frac{(HV(POF'(t)) - HV(POF^*(t)))}{HV(POF'(t))} * 100$$



Burada  $HV$  yakınsama ve çeşitliliği ölçen hiperhacim metriğidir.  $IGD$  metriği bulunan Pareto cephesinin gerçek Pareto cephesine ne kadar yakın olduğunu gösterir:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n_{POF'}} d_i^2}}{n_{POF'}}$$

Burada  $n_{POF'}$  gerçek Pareto cephesindeki çözümlerin sayısı,  $d_i$  amaç uzayında  $POF'$  un  $i$  ninci çözüm ile bulunan cephedeki en yakın çözüm arasındaki Öklit uzaklığıdır.

Ortalama  $IGD$  ( $\overline{IGD}$ ) ve ortalama  $acc$  ( $\overline{acc}$ ) değerleri sırasıyla  $t + 1$  inci değişiklikten hemen önceki zamanda hesaplanan  $IGD_t$  ve  $acc_t$  değerlerinin birikimli ortalaması olarak hesaplanır:

$$\overline{IGD} = \frac{\sum_{t=1}^{num\_of\_change} IGD_t}{num\_of\_change}, \quad \overline{acc} = \frac{\sum_{t=1}^{num\_of\_change} acc_t}{num\_of\_change}$$

**Tablo 4.1.** Dinamik çok amaçlı test problemleri

Problem	Tipi	Değişkenler	Tanımı
FDA1 [2]	Tip 1	$X_I = (x_1) \in [0,1]$ $X_{II} = (x_2, \dots, x_n) \in [-1,1]$ $n = 20$	$f_1(X_I) = x_1$ $g(X_{II}) = 1 + \sum_{x_i \in X_{II}} (x_i - G(t))^2$ $h(f_1, g) = 1 - \sqrt{f_1/g}$ $G(t) = \sin(0.5\pi t)$
FDA2 [1]	Tip 2	$X_I = (x_1) \in [0,1]$ $X_{II}, X_{III} \in [-1,1]$ $ X_{II}  = 5$ $ X_{III}  = 7$	$f_1(X_I) = x_1$ $g(X_{II}) = 1 + \sum_{x_i \in X_{II}} x_i^2$ $h(X_{III}, f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{2^{(H(t) + \sum_{x_i \in X_{III}} (x_i - H(t)/4)^2)}}$ $H(t) = 2\sin(0.5\pi(t-1))$
FDA3 [2]	Tip 2	$X_I \in [0,1]$ $ X_I  = 5$ $X_{II} \in [-1,1]$ $ X_{II}  = 25$	$f_1(X_I) = \sum_{x_i \in X_I} x_i^{F(t)}$ $g(X_{II}) = 1 + G(t) + \sum_{x_i \in X_{II}} (x_i - G(t))^2$ $G(t) =  \sin(0.5\pi t) $ $F(t) = 10^{2\sin(0.5\pi t)}$
DIMP1 [36]	Tip 1	$X_I = (x_1) \in [0,1]$ $X_{II} = (x_2, \dots, x_n) \in [-1,1]$ $n = 25$	$f_1(X_I) = x_1$ $g(X_{II}) = 1 + \sum_{x_i \in X_{II}} (x_i - G_i(t))^2$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2$ $G_i(t) = \sin\left(0.5\pi t + 2\pi\left(\frac{i}{n+1}\right)\right)^2$
DIMP2 [36]	Tip 1	$X_I = (x_1) \in [0,1]$ $X_{II} = (x_2, \dots, x_n) \in [-2,2]$ $n = 10$	$f_1(X_I) = x_1$ $g(X_{II}) = 1 + 2(n-1) + \sum_{x_i \in X_{II}} [(x_i - G(t))^2 - 2\cos(3\pi(x_i - G_i(t)))]$ $h(f_1, g) = 1 - \sqrt{f_1/g}$ $G_i(t) = \sin\left(0.5\pi t + 2\pi\left(\frac{i}{n+1}\right)\right)^2$
dMOP2 [19]	Tip 2	$x_i \in [0,1], \forall i = 1, \dots, n$ $X_I = (x_1)$ $X_{II} = (x_2, \dots, x_n)$ $n = 10$	$f_1(X_I) = x_1$ $g(X_{II}) = 1 + 9 \sum_{x_i \in X_{II}} (x_i - G(t))^2$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $G(t) =  \sin(0.5\pi t) $

			$H(t) = 0.75 \sin(0.5\pi t) + 1.25$
ZJZ [37]	Tip 2	$X_I = (x_1) \in [0,1]$ $X_{II} = (x_2, \dots, x_n) \in [-1,2]$ $n = 10$	$f_1(X_I) = x_1$ $g(X_{II}) = 1 + \sum_{x_i \in X_{II}} (x_i - G(t) - x_1^{H(t)})^2$ $h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^{H(t)}$ $G(t) = \sin(0.5\pi t)$ $H(t) = 1.5 + G(t)$

#### 4.2. Deneysel Sonuçlar (Experimental Results)

Bu bölümde, her bir deney grubunun sonuçları verilmiş ve sonuçlar değerlendirilmiştir.

##### 4.2.1. Sezgisel performans metriklerinin karşılaştırılması (Comparison of heuristic performance metrics)

Bu deney grubunda, alt seviyedeki sezgisellerin skorlarını atamak için kullanılan performans metriklerini test edilmiştir. Sezgiselleri uyguladıktan sonra, sezgisellerin skor değerleri bu performans metrikleri kullanılarak hesaplanır.

Bu çalışmada üç farklı performans metriği düşünülmüştür.  $p_1$  olarak gösterilen ilk metrik Pareto baskınlık ilişkisini kullanır (Denklem 3.6). Bu metrikte ortaya çıkan skor değeri bir önceki skordan büyükse bir gelişme vardır denir. İkinci metrik ( $p_2$ ) yine Pareto baskınlık ilişkisini kullanır. Fakat burada ortaya çıkan skor değeri belli bir eşik değerinden büyükse bir gelişme olduğu kabul edilir. Deneylerde bu eşik değeri [7] makalesinde önerildiği gibi 0.2 olarak alınmıştır. Son metrik ise *RNI* metriğini kullanır (Denklem 3.5).  $p_1$ 'de olduğu gibi sezgisellerin skorları *RNI* değerindeki değişime (iyileştirme/kötüleştirme) göre güncellenir.

Sezgisel performans metriklerini karşılaştırmak için, öğrenme içeren sezgisel seçme yönetimi olarak rastsal iniş (RD) kullanılmıştır. Tablo 4.2 farklı problemler üzerindeki ortalama *IGD* ( $\overline{IGD}$ ) ve *acc* ( $\overline{acc}$ ) değerlerini göstermektedir. Bu deneyde değişimin şiddeti  $n_t = 10$  olarak alınmıştır.

**Tablo 4.2** Farklı sezgisel performans metrikleri kullanılarak rastsal iniş tarafından üretilen ortalama *IGD* ve *acc* değerleri ( $n_t = 10$ )

Problem	$\overline{acc}$			$\overline{IGD}$		
	$p_1$	$p_2$	<i>RNI</i>	$p_1$	$p_2$	<i>RNI</i>
FDA1	<b>0.0860</b>	0.1027	0.1001	<b>0.0203</b>	0.0237	0.0231
FDA2	<b>0.0109</b>	0.0124	0.0123	<b>0.0066</b>	0.0069	0.0069
FDA3	<b>0.2838</b>	0.3286	0.3168	<b>0.0664</b>	0.0774	0.0781
DIMP1	<b>0.1666</b>	0.1893	0.1891	<b>0.0691</b>	0.0791	0.0786
DIMP2	<b>0.5448</b>	0.7229	0.6128	<b>0.2838</b>	0.3704	0.3179
dMOP2	<b>0.0118</b>	0.0130	0.0133	<b>0.0067</b>	0.0070	0.0071
ZJZ	<b>0.1395</b>	0.1498	0.1485	<b>0.0478</b>	0.0490	0.0518

Sonuçlara göre  $p_1$ , tüm test problemlerinde hem *IGD* hem de *acc* değerlerine göre en iyi sonuç veren sezgisel performans metriğidir. Bundan dolayı bundan sonraki deneylerde, sezgisel performans metriği olarak  $p_1$  kullanılacaktır.

#### 4.2.2. Sezgisel seçme yöntemlerinin karşılaştırılması (Comparison of heuristic selection methods)

Bu grupta, farklı sezgisel seçme yöntemlerinin farklı test problemleri üzerinde karşılaştırılması yapılmıştır. Daha önceden de bahsedildiği gibi, kullanılan sezgisel seçme yöntemleri şunlardır: basit rastsal (SR), rastsal iniş (RD), rastsal permütasyon (RP), rastsal permütasyon iniş (RPD), pekiştirmeli öğrenme (RL) ve Markov zinciri üst-sezgiseli (MCHH).

Kullanılan yöntemler arasındaki farkların istatistiksel olarak anlamlı olup olmadığını test etmek için tek yönlü ANOVA ve Tukey HSD testleri %95'lik bir güven aralığında gerçekleştirilmiştir. İstatistiksel karşılaştırma sonuçlarının bir özetini sağlamak için, her bir yaklaşımın farklı değişim şiddeti için tüm problemler üzerindeki diğerlerine kıyasla istatistiksel fark elde etme durumlarını sayarız. İstatistiksel karşılaştırmaların verildiği Tablo 4.3 tablosunda,  $s +$  ilgili yaklaşımın diğerlerine göre istatistiksel olarak daha iyi performans gösterdiği toplam sayısını ve  $s -$  tam tersi gösterir;  $>$  ilgili yaklaşımın diğerlerinden biraz daha iyi performans sergilediğini toplam sayısını gösterir, ancak performans farkı istatistiksel olarak anlamlı değildir ve  $<$  tam tersi gösterir. Tablo 4.3'te görüldüğü gibi pekiştirmeli öğrenme tüm problem ortamlarında her iki metriğe göre en iyi ortalama performansı sağlar. RL,  $\overline{acc}$  metriğine göre 60 durumda diğerlerinden önemli ölçüde, 24 durumda ise daha iyi sonuç vermiştir.  $\overline{IGD}$  metriğine göre ise 55 durumda diğerlerinden önemli ölçüde, 24 durumda ise daha iyi sonuç vermiştir. En iyi ikinci yaklaşım ise rastsal permütasyon iniş metodudur. Ayrıca, öğrenme içeren yaklaşımların (RD, RPD, RL ve MCHH) içermeyenlere göre (SR ve RP) daha başarılı olduğu yine Tablo 4.3'te görülmektedir.

**Tablo 4.3** Farklı sezgisel seçme yöntemleri için toplam ( $s+$ ,  $s-$ ,  $>$ ,  $<$ ) sayıları

Methods	$\overline{acc}$				$\overline{IGD}$			
	$s +$	$s -$	$>$	$<$	$s +$	$s -$	$>$	$<$
SR	3	48	6	48	4	43	13	45
RD	33	13	35	24	26	9	42	28
RP	3	47	22	33	3	39	23	40
RPD	48	7	38	12	42	7	44	12
RL	60	16	24	5	55	19	24	7
MCHH	15	31	28	31	14	27	25	39

#### 4.2.3. Literatürden seçilen yöntemlerle karşılaştırma (Comparisons to selected approaches from literature)

Bu kısımda, önerilen yaklaşım literatürde iyi bilinen DNSGAI-A ve DNSGAI-B yöntemleriyle karşılaştırılmıştır. Bir önceki grupta yapılan deneyler sonucunda sezgisel seçme yöntemi olarak RPD ve RL daha iyi sonuç verdiği için bu deney grubu için bu iki yöntem göz önüne alınmıştır.

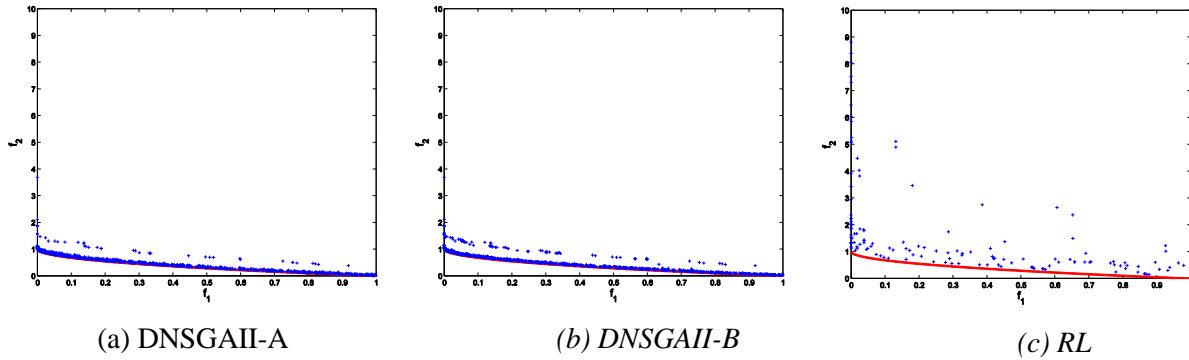
Farklı değişim şiddeti altında tüm problemler için farklı yaklaşımlar tarafından üretilen sonuçlar ( $\overline{acc}$  ve  $\overline{IGD}$ ) Tablo 4.4' de verilmiştir. Tablodaki satırlarda, farklı değişim şiddeti altında her problem için her bir yaklaşımın performansları verilmektedir. Her bir sütun ise karşılık gelen değişim şiddeti için tüm metodların performansları verilmektedir. DNSGAI-A yöntemi değişimin şiddeti çok yüksek olduğu zaman ( $n_t = 1$ ) her iki metriğe göre tüm metotlara göre daha iyi sonuç vermektedir. Değişimin şiddeti çok yüksek olduğu için ortam değiştiğinde rastgele üretilen bireylerin popülasyona katıldığı bu yöntemin başarılı olması beklenmektedir. Diğer durumlar için ise genel olarak DNSGAI-B daha iyi sonuç vermiştir. Önerilen yaklaşım FDA1, FDA3 ve ZJZ problemleri iyi sonuç vermiştir. Ayrıca, FDA3 problemi için önerilen yaklaşım en iyi sonuç veren yaklaşımdır.

Ortam değiştiğinde yaklaşımların değişimi izleme yeteneğini göstermek için elde edilen Pareto cepheler grafiklerde gösterilmiştir. Şekil 4.1 ve 4.2 de sırasıyla DIMP2 ve ZJZ problemleri için değişimin sıklığı  $\tau_t = 50$  ve şiddeti  $n_t = 10$  olduğu durumda DNSGAI-A, DNSGAI-B ve RL tarafından elde edilen

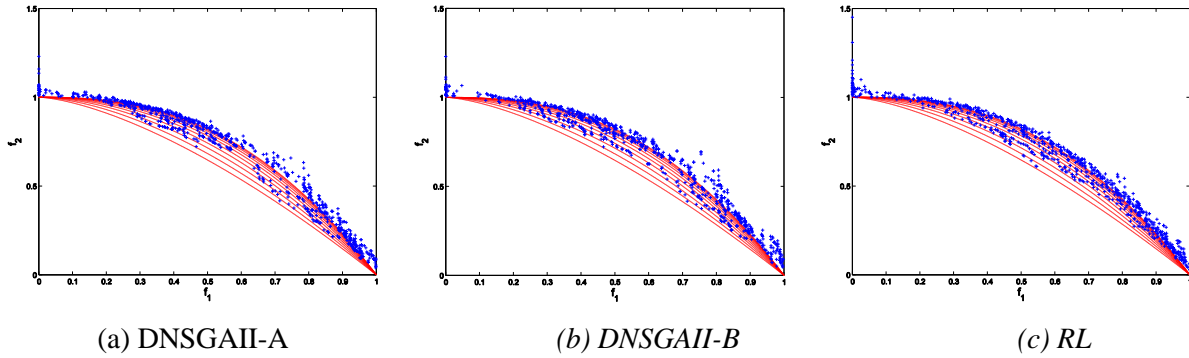
grafikler verilmiştir. Bu grafiklerdeki kırmızı çizgi gerçek POF belirtirken, mavi noktalar bulunan Pareto cephesini gösterir. DIMP2 probleminde Pareto-optimal kümesi zamanla değişirken, Pareto-optimal cephe aynı kalır. Bu problem için RL'nin izleme davranışı iyi değildir. Ayrıca, RL iyi bir yakınsama sağlayamamıştır (bkz. Şekil 4.1.). ZJZ probleminde ise hem Pareto-optimal küme hem de Pareto-optimal cephe zamanla değişir. Şekil 4.2'deki grafiklere göre RL yöntemi ZJZ problemindeki değişimleri diğer yöntemlere kıyasla daha iyi takip edebilmiştir.

**Tablo 4.4** Tüm problemler için farklı değişim şiddetlerinde farklı yöntemlerin ortalama IGD ve acc değerleri

Problem	Algoritma	$\overline{acc}$			$\overline{IGD}$		
		$n_t = 1$	$n_t = 10$	$n_t = 20$	$n_t = 1$	$n_t = 10$	$n_t = 20$
FDA1	DNSGAI-A	0.9277	0.0399	0.0375	0.2025	<b>0.0107</b>	0.0102
	DNSGAI-B	3.8034	<b>0.0399</b>	<b>0.0370</b>	0.8875	0.0108	<b>0.0102</b>
	RPD	<b>0.4685</b>	0.0818	0.0724	<b>0.0993</b>	0.0193	0.0172
	RL	1.5559	0.0717	0.0603	0.3524	0.0174	0.0150
FDA2	DNSGAI-A	0.0113	0.0040	0.0011	0.0074	<b>0.0052</b>	<b>0.0051</b>
	DNSGAI-B	<b>0.0102</b>	<b>0.0039</b>	<b>0.0010</b>	<b>0.0070</b>	0.0052	<b>0.0051</b>
	RPD	0.0198	0.0102	0.0067	0.0099	0.0064	0.0062
	RL	0.0172	0.0078	0.0043	0.0091	0.0060	0.0058
FDA3	DNSGAI-A	19.0396	0.2940	0.2282	1.6981	0.0798	0.0536
	DNSGAI-B	21.2409	0.2856	0.2241	2.7983	0.0748	0.0510
	RPD	<b>17.7067</b>	0.2639	0.2271	<b>1.5451</b>	0.0612	0.0531
	RL	18.2008	<b>0.2426</b>	<b>0.2066</b>	1.8110	<b>0.0576</b>	<b>0.0469</b>
DIMP1	DNSGAI-A	<b>0.8626</b>	0.1260	0.1070	<b>0.4408</b>	0.0565	0.0484
	DNSGAI-B	4.8835	<b>0.1117</b>	<b>0.0992</b>	2.7799	<b>0.0517</b>	<b>0.0465</b>
	RPD	1.1649	0.1440	0.1244	0.5957	0.0575	0.0507
	RL	3.5154	0.1313	0.1129	1.9753	0.0538	0.0475
DIMP2	DNSGAI-A	<b>1.1709</b>	0.1179	0.1031	<b>0.6133</b>	0.0606	0.0540
	DNSGAI-B	1.5884	<b>0.1133</b>	<b>0.1026</b>	0.8580	<b>0.0582</b>	<b>0.0536</b>
	RPD	2.1663	0.4648	0.3458	1.1311	0.2273	0.1721
	RL	1.9698	0.5212	0.2854	1.0596	0.2759	0.1514
dMOP2	DNSGAI-A	<b>0.0103</b>	0.0066	0.0064	<b>0.0063</b>	<b>0.0053</b>	<b>0.0052</b>
	DNSGAI-B	0.5775	<b>0.0059</b>	<b>0.0057</b>	0.2610	<b>0.0053</b>	<b>0.0052</b>
	RPD	0.0226	0.0112	0.0108	0.0100	0.0066	0.0065
	RL	0.2156	0.0096	0.0090	0.0945	0.0062	0.0060
ZJZ	DNSGAI-A	<b>0.4868</b>	0.2603	0.2644	0.2613	0.0785	0.0796
	DNSGAI-B	0.6603	0.2738	0.2604	0.3398	0.0835	0.0785
	RPD	0.5278	0.1419	0.1240	0.2444	<b>0.0483</b>	0.0443
	RL	0.5185	<b>0.1391</b>	<b>0.1228</b>	<b>0.2435</b>	0.0489	<b>0.0441</b>



**Şekil 4.1** DIMP2 problemi üzerinde  $\tau_t = 50$  ve  $n_t = 10$  değerleri için (a) DNSGAII-A, (b) DNSGAII-B ve (c) RL tarafından elde edilen Pareto cepheler.



**Şekil 4.2** ZJZ problemi üzerinde  $\tau_t = 50$  ve  $n_t = 10$  değerleri için (a) DNSGAII-A, (b) DNSGAII-B ve (c) RL tarafından elde edilen Pareto cepheler.

## 5. SONUÇ (CONCLUSION)

Bu çalışmada çok amaçlı eniyileme problemlerini çözmek için geliştirilen ve evrimsel algoritmaları ve üst-sezgisellerin birlikte kullanıldığı genel bir çerçevenin performansı incelenmiştir. Yapı içinde evrimsel algoritma olarak literatürde iyi sonuçlar veren NSGA-II kullanılmıştır. Önerilen yöntemin performansı farklı sezgisel seçme yöntemleri kullanılarak test edilmiştir. Deneysel sonuçlara göre öğrenme içeren sezgisel seçme yöntemlerinin diğerlerine oranla daha iyi sonuç verdiği gözlemlenmiştir. Önerilen yapının başarımı literatürde iyi bilinen iki yöntemle karşılaştırılmıştır. Yapılan deneyler sonucunda bazı problemler (FDA3 ve ZJZ) için önerilen yapının diğer yöntemlere göre iyi sonuç verdiği, genel olarak ise bu yöntemlerle rekabet edebilir seviyede iyi sonuçlar verdiği gözlemlenmiştir. Bunun sonucunda yapılacak sonraki çalışmalarda, yapı içinde farklı evrimsel algoritma tiplerinin performansı incelenecektir.

## TEŞEKKÜR (ACKNOWLEDGMENTS)

Yazar TÜBİTAK-2219 Yurt Dışı Doktora Sonrası Burs Programı vasıtasıyla destekleri için TÜBİTAK' a teşekkür eder. Yazar ayrıca Prof. Dr. Kalyanmoy Deb' e çok amaçlı eniyileme ile ilgili yardımlarından dolayı teşekkür eder.

**KAYNAKLAR (REFERENCES)**

- [1] Deb, K., Rao N., U.B., ve Karthik, S.: "Dynamic Multi-objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-thermal Power Scheduling", **Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings**, in Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., ve Murata, T. (Ed.)^(Eds.), Springer Berlin Heidelberg, 803-817, 2007.
- [2] Farina, M., Deb, K., ve Amato, P.: "Dynamic Multiobjective Optimization Problems: Test Cases, Approximation, and Applications", **Evolutionary Multi-Criterion Optimization: Second International Conference, EMO 2003, Faro, Portugal, April 8–11, 2003. Proceedings**, in Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., ve Deb, K. (Ed.)^(Eds.), Springer Berlin Heidelberg, 311-326, 2003.
- [3] Helbig, M., "Solving dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation", University of Pretoria, Faculty of Engineering, Built Environment and Information Technology, 2012.
- [4] Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., ve Qu, R., "Hyper-heuristics: a survey of the state of the art", **Journal of the Operational Research Society**, Cilt 64, No 12, 1695-1724, 2013.
- [5] Özcan, E., Bilgin, B., ve Korkmaz, E.E., "A comprehensive analysis of hyper-heuristics", **Intell. Data Anal.**, Cilt 12, No 1, 3-23, 2008.
- [6] Cowling, P.I., Kendall, G., ve Soubeiga, E., "A Hyperheuristic Approach to Scheduling a Sales Summit". **Proc. Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III**, 176-190, 2001.
- [7] McClymont, K., Keedwell, E., Savić, D., ve Randall-Smith, M., "A general multi-objective hyper-heuristic for water distribution network design with discolouration risk", **Journal of Hydroinformatics**, Cilt 15, No 3, 700-716, 2013.
- [8] Kiraz, B., Etaner-Uyar, A.Ş., ve Özcan, E., "Selection hyper-heuristics in dynamic environments", **Journal of the Operational Research Society**, Cilt 64, No 12, 1753-1769, 2013.
- [9] Deb, K., Pratap, A., Agarwal, S., ve Meyarivan, T., "A fast and elitist multiobjective genetic algorithm: NSGA-II", **IEEE Transactions on Evolutionary Computation**, Cilt 6, No 2, 182-197, 2002.
- [10] Coello, C.A., "An updated survey of GA-based multiobjective optimization techniques", **ACM Comput. Surv.**, Cilt 32, No 2, 109-143, 2000.
- [11] Jin, Y., ve Branke, J., "Evolutionary optimization in uncertain environments-a survey", **IEEE Transactions on Evolutionary Computation**, Cilt 9, No 3, 303-317, 2005.
- [12] Yang, S., ve Yao, X., "**Evolutionary Computation for Dynamic Optimization Problems**", 2013.
- [13] Cobb, H.G., "An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments", Rep.No, Naval Research Lab., Washington, DC, 1990.
- [14] Uyar, A.Ş., ve Harmanci, A.E., "A new population based adaptive domination change mechanism for diploid genetic algorithms in dynamic environments", **Soft Computing**, Cilt 9, No 11, 803-814, 2005.

- [15] Yang, S., ve Yao, X., "Population-Based Incremental Learning With Associative Memory for Dynamic Environments", **IEEE Transactions on Evolutionary Computation**, Cilt 12, No 5, 542-561, 2008.
- [16] Yang, S., "Genetic algorithms with memory-and elitism-based immigrants in dynamic environments", **Evol. Comput.**, Cilt 16, No 3, 385-416, 2008.
- [17] Wang, Y., ve Li, B., "Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment". **Proc. 2009 IEEE Congress on Evolutionary Computation**, 630-637, 18-21 May 2009, 2009.
- [18] Branke, J., "**Evolutionary Optimization in Dynamic Environments**", Kluwer Academic Publishers, 2001.
- [19] Goh, C.-K., ve Tan, K.C., "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization", **Trans. Evol. Comp**, Cilt 13, No 1, 103-127, 2009.
- [20] Sahmoud, S., ve Topcuoglu, H.R., "A Memory-Based {NSGA-II} Algorithm for Dynamic Multi-objective Optimization Problems". **Proc. 19th European Conference, EvoApplications 2016**, Porto, Portugal, 296--310, 2016.
- [21] Helbig, M., Deb, K., ve Engelbrecht, A.P., "Key challenges and future directions of dynamic multi-objective optimisation". **Proc. {IEEE} Congress on Evolutionary Computation**, Vancouver, BC, Canada, 1256--1261, 2016.
- [22] Nareyek, A.: "Choosing Search Heuristics by Non-Stationary Reinforcement Learning", **Metaheuristics: Computer Decision-Making**, in (Ed.)^(Eds.), Springer US, 523-544, 2004.
- [23] Ozcan, E., Uyar, S.E., ve Burke, E., "A greedy hyper-heuristic in dynamic environments". **Proc. Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers**, Montreal, Canada, 2201-2204, 2009.
- [24] Uludağ, G., Kiraz, B., Etaner-Uyar, A.Ş., ve Özcan, E., "A hybrid multi-population framework for dynamic environments combining online and offline learning", **Soft Computing**, Cilt 17, No 12, 2327-2348, 2013.
- [25] Topcuoglu, H.R., Ucar, A., ve Altin, L., "A hyper-heuristic based framework for dynamic optimization problems", **Applied Soft Computing**, Cilt 19, No, 236-251, 2014.
- [26] Burke, E.K., Silva, J.D.L., ve Soubeiga, E.: "Multi-Objective Hyper-Heuristic Approaches for Space Allocation and Timetabling", **Metaheuristics: Progress as Real Problem Solvers**, in Ibaraki, T., Nonobe, K., ve Yagiura, M. (Ed.)^(Eds.), Springer US, 129-158, 2005.
- [27] McClymont, K., ve Keedwell, E.C., "Markov chain hyper-heuristic (MCHH): an online selective hyper-heuristic for multi-objective continuous problems". **Proc. Proceedings of the 13th annual conference on Genetic and evolutionary computation**, Dublin, Ireland, 2003-2010, 2011.
- [28] Zitzler, E., Laumanns, M., Thiele, L., "SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm", Rep.No: 103, Swiss Federal Institute of Technology (ETH) Zurich 2001.
- [29] Gomez, J.C., ve Terashima-Marín, H.: "Approximating Multi-Objective Hyper-Heuristics for Solving 2D Irregular Cutting Stock Problems", **Advances in Soft Computing: 9th Mexican International Conference on Artificial Intelligence, MICAI 2010, Pachuca, Mexico, November 8-13, 2010, Proceedings, Part II**, in Sidorov, G., Hernández Aguirre, A., ve Reyes García, C.A. (Ed.)^(Eds.), Springer Berlin Heidelberg, 349-360, 2010.

- [30] Kumari, A.C., Srinivas, K., ve Gupta, M.P., "Software module clustering using a hyper-heuristic based multi-objective genetic algorithm". **Proc. 2013 3rd IEEE International Advance Computing Conference (IACC)**, 813-818, 22-23 Feb. 2013, 2013.
- [31] Suganthan, P.N., "Performance assessment on multi-objective optimization algorithms". **Proc. IEEE Conference on Evolutionary Computation Special Session-competition on performance assessment of multi-objective optimization algorithms**, 2007.
- [32] Das, S., ve Suganthan, P.N., "Differential Evolution: A Survey of the State-of-the-Art", **IEEE Transactions on Evolutionary Computation**, Cilt 15, No 1, 4-31, 2011.
- [33] Tan, K.C., Lee, T.H., ve Khor, E.F., "Evolutionary Algorithms for Multi-Objective Optimization: Performance Assessments and Comparisons", **Artificial Intelligence Review**, Cilt 17, No 4, 251-290, 2002.
- [34] Özcan, E., Misir, M., Ochoa, G., ve Burke, E.K., "A Reinforcement Learning-Great-Deluge Hyper-Heuristic for Examination Timetabling", **Int. J. Appl. Metaheuristic Comput.**, Cilt 1, No 1, 39-59, 2010.
- [35] Zhang, Q., Zhou, A., ve Jin, Y., "RM-MEDA: A Regularity Model-Based Multiobjective Estimation of Distribution Algorithm", **IEEE Transactions on Evolutionary Computation**, Cilt 12, No 1, 41-63, 2008.
- [36] Koo, W.T., Goh, C.K., ve Tan, K.C., "A predictive gradient strategy for multiobjective evolutionary algorithms in a fast changing environment", **Memetic Computing**, Cilt 2, No 2, 87-110, 2010.
- [37] Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., ve Tsang, E., "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization". **Proc. Proceedings of the 4th international conference on Evolutionary multi-criterion optimization**, Matsushima, Japan, 832-846, 2007.