

# Kalıcı Durum Evrimsel Algoritmada Yerine Koyma Tekniklerinin Deneysel İncelenmesi

Alper Çiftçi<sup>1</sup> Şima Etaner-Uyar<sup>2</sup>

Bilgisayar Mühendisliği Bölümü, İstanbul Teknik Üniversitesi, İstanbul  
ciftcial@itu.edu.tr<sup>1</sup>, etaner@cs.itu.edu.tr<sup>2</sup>

## Özet

*Kalıcı durum evrimsel algoritmada genellikle iki ebeveyn bireyden yeni bir birey üretilmekte ve toplumdaki eski bir bireyin yerine konulmaktadır. Bu noktada nasıl bir yöntemin kullanılacağı toplumdaki bireylerin başarılarının iyileşmesi açısından önem taşımaktadır. Bu amaçla literatürde çeşitli yöntemler önerilmiştir. Bu çalışmada bunlara ek olarak, tepetürmanmalı en kötünün yerine koyma (replace worst-climb) yöntemi önerilmiştir. Bu çalışmada yerine koyma yöntemlerinin başarıları deneysel olarak incelenmiş ve karşılaştırılmıştır. Bu yöntemlerin test edilmesinde literatürde yaygın olarak kullanılan ve birbirinden farklı özellik gösteren bazı test problemleri kullanılmıştır.*

KDEA modeline göre, bir sonraki toplumu oluşturmak için mevcut toplumdaki birkaç birey değiştirilir. Böylece KDEA modelinde nesilden nesile toplumda çok fazla değişim gözlenmez.[2] Şekil 3.1'de KDEA'daki temel adımlar gösterilmiştir.

- (1) Rasgele ilk toplumu oluştur
- (2) İki ebeveyn seç
- (3) Çaprazlama ile yeni birey üret
- (4) Yeni bireyi mutasyona uğrat
- (5) Seçilen bireyin yerine yeni bireyi koy
- (6) Durma şartı sağlanmadıysa (2)'ye git
- (7) En iyi çözümü döndür

## 1. Giriş

Toplum modellerine göre iki farklı evrimsel algoritmada (EA) bahsedilebilir. Bunlardan birincisi her nesilde toplumdaki bireylerin tamamıyla değiştiği ve yerine yeni bireylerin geldiği nesilsel (generational), ikincisi ise toplumdan çok az birey seçilip bunlardan yeni birey oluşturmak suretiyle toplumun nesilden nesile çok fazla değişmeyen bir karakteristik gösterdiği kalıcı durum (steady state) EA' dır. [6]

Kalıcı durum evrimsel algoritmada (KDEA) toplumun her nesilde tamamen değişmemesi yalnızca çok küçük bir kısmının değişmesi nedeni ile hangi yerine koyma yönteminin uygulandığı önemlidir. Bu amaçla çeşitli yöntemler kullanılmaktadır. Bu çalışmada, yerine koyma yöntemleri üzerinde bazı test fonksiyonları kullanılarak deneyler yapılmış ve tepetürmanmanın kullanıldığı yeni bir yöntem önerilmiştir.

Bölüm 2'de kalıcı durum evrimsel algoritma ve yerine koyma yöntemlerinden bahsedilmiştir. Bölüm 3'te çalışmada kullanılan test problemleri anlatılmış ve bölüm 4'te de deneysel sonuçlara yer verilmiştir. Bölüm 5'te genel yorumlar ve olası geliştirmeler sunulmuştur.

## 2. Kalıcı Durum Evrimsel Algoritma

EA'lar arama uzayında her biri bir nokta olan bireylerden oluşan bir toplum üzerinde çalışır. Bireylerin kalitesi yani problemin çözümüne yakınlığı bir başarı fonksiyonu ile ölçülür. EA mevcut bireylerden çeşitli yöntemlerle yeni bireyler üreterek toplumu iyileştirmeye ve çözüme yaklaştırmaya çalışır.

Şekil 1. Kalıcı durum evrimsel algoritma

Algoritmada ilk adımda N birey içeren rasgele bir toplum oluşturulmaktadır. Burada toplum oluşturulurken aynı kromozoma sahip, birbirinin aynısı iki birey oluşturulmamaya dikkat edilir (duplicate elimination). 2. adımda toplumdaki bireyler değerlendirilip çocuk bireyi oluşturmak üzere en uygun iki birey seçilir. Bir sonraki adımlarda çaprazlaşma ve mutasyon ile yeni bir birey oluşturulur. 5. adımda yeni birey toplumdaki çıkarılmak üzere seçilen diğer bir bireyin yerine topluma katılır. Topluma girecek bireyin genotipinin toplumda var olan bir bireyle aynı olmaması gerekmektedir. Bu döngü sonlanma koşulu sağlanmadığı sürece devam eder. KDEA ile ilgili detaylı bilgiye [1]'den ulaşılabilir.

Bu çalışmada turnuva seçim (*tournament selection*) yöntemi, tekbiçimli çaprazlama (*uniform crossover*) ve bit-bazlı mutasyon kullanılmıştır.

Turnuva seçim yönteminde toplumdan  $k$  tane bireyden oluşan rasgele bir grup seçildikten sonra bu grubun en iyi bireyi seçilir. Toplumdan  $n$  tane birey seçmek için bu işlem  $n$  defa tekrarlanır. Bu çalışmada turnuva  $k=2$  olarak alınmıştır.

Tekbiçimli çaprazlamada yeni birey oluşturulurken, bireyin kromozomundaki her gen eşit olasılıkla ya ilk bireyden ya da ikinci bireyden alınır. Ayrıca çaprazlamada  $p_c$  çaprazlama oranına göre çaprazlama uygulanıp uygulanmayacağına karar verilir. Bu çalışmada çaprazlama oranı  $p_c=1$  olarak alınmıştır.

Mutasyon bireyin genlerinin farklılaştırılmasıdır. Bireyin genlerine belirli bir  $p_m$  mutasyon oranı ile uygulanır. Bu oran genellikle çok küçüktür ve sadece birkaç gende değişim olmasını sağlar. Mutasyona

uğrayan genin değeri değiştirilir. Bu çalışmada mutasyon oranı  $p_m=1/kromozom\ boyu$  olarak alınmıştır.

## 2.1 Yerine Koyma Aşamasi

Yerine koyma KDEA için önemli bir aşamadır. Genetik işlemler sonucunda üretilen birey toplumdaki bir bireyin yerini alacaktır. Bu aşamada yeni bireyin hangi bireyin yerine konulacağı konusunda önerilen yöntemler toplumdaki çıkarılacak bireyin seçiminde rasgele, yaşa göre ve başarıma göre değerlendirme yapmaktadır.

**2.4.1 Rasgele bir bireyin yerine koyma.** Bu yöntemde rasgele seçilen bir birey toplumdaki çıkarılır ve bu çıkarılan bireyin yerine yeni birey topluma girer [5].

**2.4.2 En iyi birey hariç rasgele bir bireyin yerine koyma.** Bu yöntemde toplumdaki mevcut en iyi birey hariç, rasgele seçilen bir birey toplumdaki çıkarılır ve bu çıkarılan bireyin yerine yeni birey topluma girer [5].

**2.4.3 En yaşlının yerine koyma.** Bu yöntemde oluşan yeni birey toplumdaki yaşı en büyük olan bireyin yerine konulur [5]. İlk toplumdaki tüm bireylerin yaşı sıfırdır. İzleyen her nesilde toplumdaki tüm bireylerin yaşı bir artırılır. Topluma giren her yeni bireyin yaşı sıfır olur.

**2.4.4 En iyi birey hariç en yaşlının yerine koyma.** Bu yöntemde de oluşan yeni birey toplumdaki mevcut en iyi bireyin dışında yaşı en büyük olan bireyin yerine konulur [5]. En yaşlı bireyin en iyi olduğu durumda bir sonraki yaşlı birey seçilir.

**2.4.5 En benzerinin yerine koyma.** Bu yöntemde oluşan yeni birey toplumdaki kendisine Hamming uzaklığı olarak en yakın olan bireyin yerine konulur. Böylece en benzer birey ile yer değiştirilmiş olur.

**2.4.6 En iyi birey hariç en benzerinin yerine koyma.** Bu yöntemde oluşan yeni birey toplumdaki mevcut en iyi bireyin dışında kendisine Hamming uzaklığı olarak en yakın olan bireyin yerine konulur. En benzer bireyin en iyi olduğu durumda bir sonraki benzer birey seçilir.

**2.4.7 En kötünün yerine koyma.** Bu yöntemde yeni birey toplumdaki başarıma değeri en kötü olan birey ile yer değiştirilir [5]. Bu amaçla ilk olarak en kötü birey bulunur. Daha sonra yeni bireyin başarıma en kötü bireyin başarıma ile karşılaştırılır ve yeni birey daha iyi ise diğer bireyin yerini alır.

**2.4.8 Tepe-tırmanmalı en kötünün yerine koyma.** En kötünün yerine koymada yeni birey en kötünden iyi değilse topluma alınmaz ama bu birey en kötünün üzerinde bulunduğu tepeden daha yüksek bir tepede olabilir. Bu ise yeni bireyin potansiyel olarak daha iyi bir çözüm olduğu anlamına gelir. En kötünün yerine Tepe-tırmanmalı yöntemde yeni bireyi en kötü bireyle karşılaştırıp yerine koymak yerine, tepe tırmandırarak en kötü bireyin başarıma ulaşımını ulaşıp ulaşmadığına

bakılır. Tırmanma sonucu yeni birey en kötü bireyin başarıma ulaşımını ulaşıyor ise en kötü birey toplumdaki çıkarılır ve yeni birey onun yerine topluma girer. Tepe-tırmanmalı yöntemin iki farklı yaklaşımı mevcuttur. *Lamarck* yaklaşımına göre yeni birey tırmandırıldıktan sonra en kötü bireyin başarıma ulaşımına ulaşımına genotipinin mevcut haliyle topluma girer [1]. *Darwin* yaklaşımına göre ise tırmanan birey en kötü bireyin başarıma ulaşımına ulaşınca tırmanma öncesi genotipinin haliyle topluma girer [1].

Çalışmada tepe-tırmanma için *en dik iniş (steepest descent)* [13] yöntemi kullanılmıştır. Bu yöntemde bireyin tüm komşularına bakılıp en iyisi seçilerek tırmanma devam etmektedir.

## 3. Test Problemleri

Çalışmada yerine koyma yöntemlerini sınamak ve karşılaştırabilmek amacıyla yaygın bazı test fonksiyonları ve problemler kullanılmıştır. Bununla birlikte farklı özellikler gösteren problemler seçmeye özen gösterilmiştir. Seçilen problemler aşağıda açıklanmıştır.

### 3.1 Onemax Problemi

Onemax problemi tek bir global maksimumu olan tek tepeli bir fonksiyondur. Onemax probleminde amaç yalnızca 0 ve 1'lerden oluşan bir dizide 1'lerin sayısını enbüyüklemektir. Burada birey için başarıma, sahip olduğu birlerin sayısıdır.

### 3.2 Shekel's Foxholes (f5)

Foxholes fonksiyonunda bir çok yerel optimum mevcuttur. Standart algoritmalar için bu yerel optimumlar sorun teşkil etmektedir [7]. Testler için bu fonksiyonun 25 yerel optimum olan aşağıdaki şekli kullanılmıştır [9].

$$f_5(\vec{x}) = \frac{1}{K} + \sum_{j=1}^{25} \frac{1}{c_j + \sum_{i=1}^2 (x_i - a_{ij})^6}$$

$$(a_{ij}) = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 \end{pmatrix}$$

$$K = 500; f_5(a_{ij}, a_{2j}) \approx c_j = j;$$

$$-65.536 \leq x_i \leq 65.536$$

$$\min(f_5) = f_5(-32, -32) \approx 1$$

### 3.3 Rastrigin Fonksiyonu (f7)

Rastrigin'in fonksiyonu (f7) çok geniş bir arama uzayı ve çok fazla yerel minimum olması nedeniyle EA'lar için oldukça zordur. Fonksiyonun yüzeyi modülasyonun genişliğini ve frekansını kontrol eden A ve w değişkenleri ile belirlenir [7]. Fonksiyonun minimum değeri  $x_i = 0, i = 1, \dots, n$  için  $f(x) = 0$ 'dır [9].

$$f_7(\vec{x}) = nA + \sum_{i=1}^n x_i^2 - A \cos(wx_i)$$

$$A = 10 ; w = 2\pi ; -5.12 \leq x_i \leq 5.12$$

$$\min(f_7) = f_7(0, \dots, 0) = 0 \quad [9]$$

### 3.4 Çok Boyutlu Çanta Problemi

Çok boyutlu çanta problemi basit 0-1 çanta probleminin genel bir halidir. Bu problemde amaç çantaların kısıtlarını sağlayacak ve seçilen nesnelerin toplam faydasını enbüyükleyecek şekilde nesnelere seçmektir [10]. Problemin matematiksel ifadesi şöyledir;

$$\sum_{j=1}^n p_j x_j \text{ değerini}$$

$$\sum_{j=1}^n w_{ij} x_j \leq c_i \quad i = 1, \dots, d,$$

$x_j \in \{0,1\}$ ,  $j = 1, \dots, n$ . olacak şekilde enbüyükle.

Burada  $n$  nesnelerin sayısı,  $d$  çanta sayısı,  $c_i$  çanta  $i$ 'nin kapasitesi,  $p_j$   $j$ . nesnenin faydası,  $x_j$   $j$ . nesnenin alınıp alınmadığını gösteren değişken,  $w_{ij}$   $j$ . nesnenin  $i$ . çantadaki ağırlığıdır.

Çok boyutlu çanta probleminin bu çalışmadaki diğer problemlerden farkı, kısıtlar nedeniyle geçerli olmayan çözümlerin de bulunmasıdır. Yani toplumda geçerli olmayan, çözüm temsil etmeyen bireyler de bulunmaktadır. En iyi başarıma sahip bireyler geçerli çözümler bölgesinin sınırına yakın olanlardır [2]. Ayrıca çok boyutlu çanta probleminde sıkılık oranı (tightness ratio)  $\alpha_i$  önem taşımaktadır. Sıkılık oranı bir çanta için çanta kapasitesinin tüm nesnelerin o çantadaki ağırlıklarına oranıdır. Bu oranların en küçüğüne en düşük sıkılık oranı (minimum tightness ratio) denir [2].

$$\alpha_i = \sum_{j \in J} r_{ij} / c_i \quad i \in I = \{1, \dots, m\}$$

$$\alpha_{\min} = \min\{\alpha_i \mid i \in I\}$$

Çalışmada örnek test problemi olarak "OR-Library" deki *mknapl* veri dosyası kullanılmıştır [11]. Bu test dosyası içinde 7 tane problem vardır. Bu çalışmada 28 nesne ve 10 çantanın bulunduğu 5. problem kullanılmıştır. Minimum sıkılık oranı 0.609 dur.

### 3.5 Çok Tepeli Fonksiyonlar

Çalışmada tepe yükseklikleri eşit olan çok tepeli fonksiyon elde etmek amacıyla [12]'deki çok tepeli fonksiyon üretici kullanılmıştır. Bu üretici belirlenen kromozom boyunda 0-1 dizileri üretmektedir. Bu diziler fonksiyonun tepe noktalarını temsil etmektedir. Ayrıca üretilmek istenen tepe noktası sayısı da belirlenebilmektedir. Başarım, kromozomdaki en yakın tepe noktasıyla ortak olan bitlerin sayısıdır. Bu üretici ile kromozom uzunluğunun 100 olduğu eşit yükseklikli 5 ve 10 tane tepeye sahip iki fonksiyon

üretilmiş ve çalışmada bu fonksiyonlar da kullanılmıştır. Onemax hariç diğer fonksiyonlar da çok tepeli olmalarına karşın, bu bölümdeki problemlerin özelliği tüm tepelerin eşit yükseklikte ve özellikte olmalarıdır.

## 4. Deneysel Sonuçlar

Yerine koyma yöntemlerinin testlerinde algoritmanın durma koşulu 10000 başarımla hesaplanarak belirlenmiştir. Toplum boyu 50, çaprazlama oranı 1, mutasyon oranı (1/kromozom boyu) olarak alınmıştır. Her problem ve her yöntem için algoritma 100 defa çalıştırılmış ve her başarımla hesaplanarak mevcut en iyi bireyin başarımla ortalamaları elde edilmiştir. Kromozomların kodlanmasında gray kodlama [1] kullanılmıştır.

Her problem için elde edilen sonuçlar 0-5000 başarımla hesaplanarak çizdirilmiştir. Grafiklerde kullanılan kısaltmalar Çizelge-6.1'de verilmiştir.

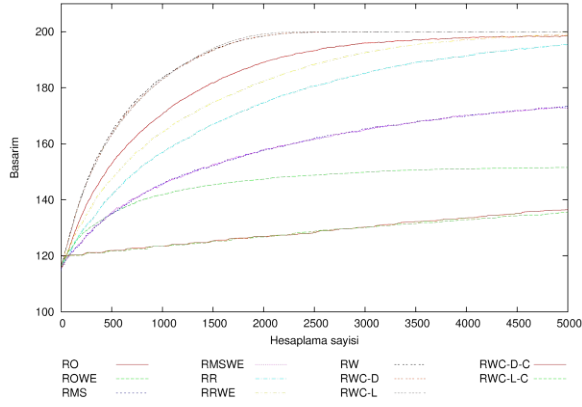
Çizelge 6.1: Yerine Koyma yöntemleri

Kısaltma	Yerine Koyma Yöntemi
RO	En yaşlının yerine koyma
ROWE	En iyi birey hariç en yaşlının yerine koyma
RMS	En benzerinin yerine koyma
RMSWE	En iyi birey hariç en benzerinin yerine koyma
RR	Rasgele bir bireyin yerine koyma
RRWE	En iyi birey hariç rasgele bireyin yerine koyma
RW	En kötünün yerine koyma
RWC-D	Tepe-tırmanmalı en kötünün yerine koyma - Darwinsel
RWC-L	Tepe-tırmanmalı en kötünün yerine koyma - Lamarcksal
RWC-D-C	Tepe-tırmanmalı en kötünün yerine koyma - Darwinsel - Tırmanma başarımla hesapları dahil
RWC-L-C	Tepe-tırmanmalı en kötünün yerine koyma - Lamarcksal - Tırmanma başarımla hesapları dahil

### 4.1 Onemax Problemi

Onemax probleminde kromozom boyu 200 seçilmiştir. En iyi başarımla 200 olacaktır. Bu problem için EKYK, Darwinsel ve Lamarcksal tepe-tırmanmalı EKYK yöntemleri en iyi ve benzer sonuçları çıkarmıştır. Bunlardan sonra en iyi performansı EYYK göstermiştir. EYYK ise bu problem için çok iyi bir performans göstermemiştir. Daha sonra ERBYK yöntemi gelmektedir. Tek başına RBYK ise ERBYK yöntemine göre daha kötü bir sonuç çıkarmıştır. EBYK yöntemi ise en birey tutulsun ya da tutulmasın benzer performans göstermiştir. Tırmanma sırasında başarımla hesapları sayıldığında ise tepe-tırmanmalı EKYK yöntemleri kromozom boyunun büyük olması nedeni ile iyi performans göstermemiştir.

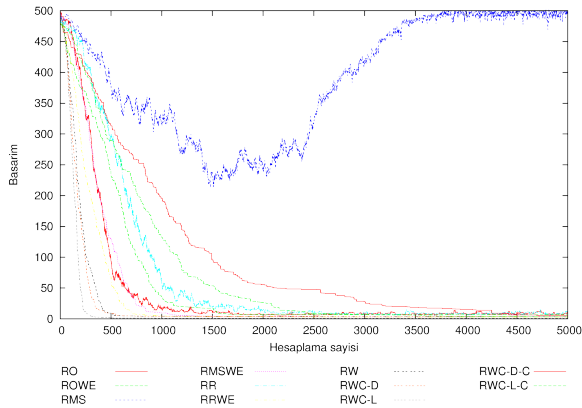
Onemax probleminin tek tepeli bir problem olması nedeni ile tepe-tırmanmalı yöntemlerin bu problemde daha iyi bir sonuç göstermesi beklenen bir durum değildi.



Şekil 6.1: Onemax sonuçları

#### 4.2 Shekel's Foxholes (f5) fonksiyonu

F5 fonksiyonu için en iyi sonucu Tepe-tırmanmalı EKYK - Lamarcksal yöntem çıkarmıştır. Bunda tırmanma sırasındaki başarımların dahil edilmemesi etkili olmuştur. Başarımların maliyetli olduğu durumlarda tepe-tırmanmalı yöntemler etkili olmamaktadır. Bununla birlikte bu yöntem Tepe-tırmanmalı EKYK - Darwinsel yöntemden daha iyi bir performans sergilemektedir. Tırmanma sırasındaki başarımların dahil edildiğinde ise tepe-tırmanmalı yöntem EKYK yöntemine göre daha iyi olmamakla birlikte, burada da Lamarck yaklaşımının Darwin yaklaşımına göre daha iyi olduğu görülmektedir. EYYK en iyi birey tutulmadığı durumda tutulduğu duruma göre daha fazla iyileşme sağlamaktadır. EBYK'da ise en iyi birey tutulmadığında bir süre iyileşme olduktan sonra iyi bireyler kaybedilmeye başlayınca başarımlar gitgide kötü hale gelmektedir. RBYK ise en iyi bireyin tutulduğu durumda daha başarılı olmaktadır.

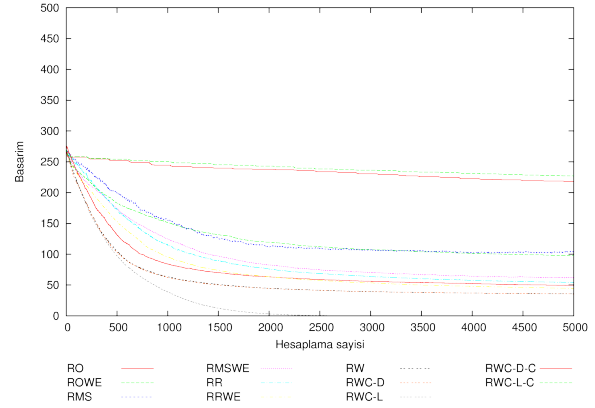


Şekil 6.2: f5 fonksiyonu sonuçları

#### 4.3 Rastrigin Fonksiyonu (f7)

Bu problem için de en iyi başarımların artışı tırmanma sırasındaki başarımların dahil edilmediğinde tepe-tırmanmalı EKYK Lamarck yaklaşımı ile sağlamıştır. Yine bu yöntem tepe-tırmanmalı EKYK - Darwinsel yöntemden daha iyi bir performans sergilemektedir. Tırmanma sırasındaki başarımların dahil edildiğinde ise tepe-tırmanmalı yöntem büyük kromozom boyu nedeni ile

başarımların iyileştirilmesi sağlayamamaktadır. Rastrigin fonksiyonunda RBYK, EYYK, EBYK yöntemleri f5'e göre daha düzgün bir başarımların iyileşmesi sağlamaktadır. Yine f7'de de EYYK en iyi birey tutulmadığı durumda tutulduğu duruma göre daha iyi bir iyileşme sağlamaktadır. Bu fonksiyon için EBYK f5'deki gibi en iyi birey tutulmadığında bir süre iyileşme olduktan sonra iyi bireyler kaybedilmeye ve başarımlar gitgide kötü hale gelmektedir. Fakat bu süreç daha uzun sürmektedir. RBYK ise en iyi bireyin tutulduğu durumda daha başarılı olmaktadır.

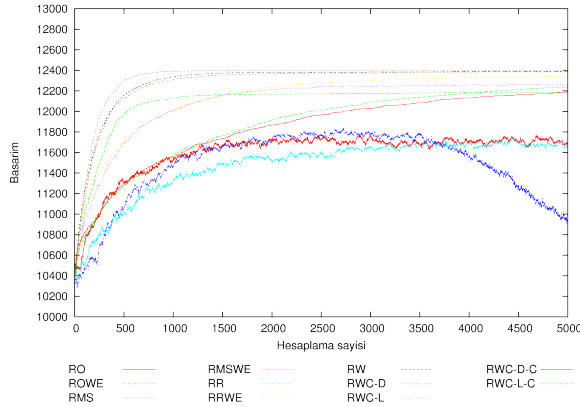


Şekil 6.3: f7 fonksiyonu sonuçları

#### 4.4 Çok Boyutlu Çanta Problemi

Çok boyutlu çanta probleminin çözümü uygun çözümler bölgesinin sınırına yakın olmaktadır ve başarımların hesabında kullanılan ceza fonksiyonu toplumu bu sınıra itmektir [2]. Geçerli ve geçersiz çözümler bu sınırdaki komşudur. İlk toplumda birçok geçersiz çözümler bulunmaktadır. EBYK yöntemi kullanıldığında bir süre sonra toplum sınıra yaklaştıkça üretilen geçersiz yeni bireyler kendilerine komşu olan geçerli bireylerin yerini almaya başlamaktadırlar. Bu yüzden en iyi başarımlar düşmeye başlamaktadır. Sıkılık oranları düşük olduğunda ilk toplumda çok miktarda geçersiz çözümler bulunmaktadır. Algoritma ilerledikçe daha fazla geçerli çözümler elde edilmektedir ve yaşlı bireyler geçerli çözümler olmamaktadırlar. Sürekli en yaşlı ile yer değiştirildiğinde, geçersiz bireyler yok olmakta ve en iyi başarımlar artmaktadır. Fakat zamanla toplum geçerli çözümler sınırının yakınlarına yakınsamaktadır. Bu durumda neredeyse tüm bireyler geçerli ve birbirilerine benzer olduğundan, en yaşlı ile yer değiştirmek artık iyileşme sağlamamakta ve en iyi başarımlar belirli bir düzeyde kalmaktadır. RBYK ise genel olarak EYYK ile benzerlik göstermektedir. Bununla birlikte EYYK'da başlangıçta en yaşlı bireyler geçersiz çözümler olduğundan yer değiştirmelerde geçersiz çözümler toplumdaki çıkarılmaktadırlar. RBYK'da ise geçerli çözümlerle de yer değiştirme olacağından başlangıçta başarımların artışı daha düşüktür. Tepe-tırmanmalı EKYK Lamarck yaklaşımı ile en iyi başarımların artışı sağlamaktadır. Burada yine tırmanma sırasındaki başarımların dahil edilmemesi etkili olmaktadır. Bununla birlikte bu yaklaşım Darwin yaklaşımına göre daha iyi bir performans sergilemektedir. Tırmanma sırasındaki başarımların dahil edildiğinde ise

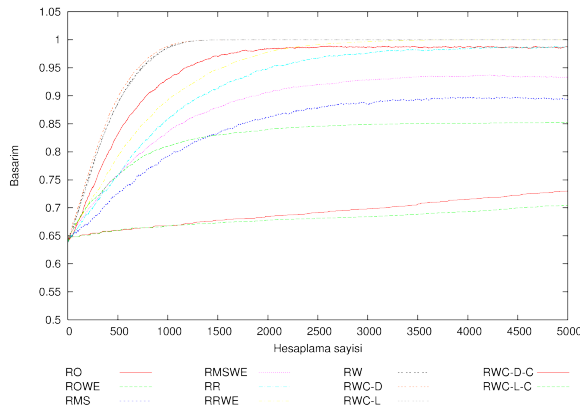
tepe-tırmanmalı yöntem EKYK yöntemine göre daha iyi olmamakla birlikte, burada da Lamarck yaklaşımının Darwin yaklaşımına göre daha iyi olduğu görülmektedir.



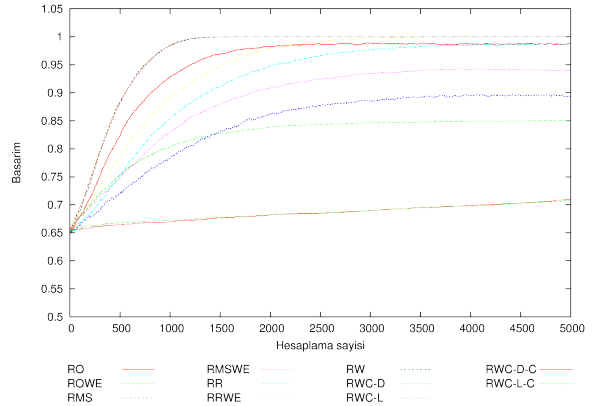
Şekil 6.4: Çok boyutlu çanta problemi sonuçları

#### 4.5 Çok Tepeli Fonksiyonlar

Biri 5 diğeri 10 tepeli olan iki fonksiyon kromozom boyu 100 alınarak sınanmıştır. Her iki fonksiyon içinde sonuçlar birbirine yakın çıkmıştır. Onemax probleminde olduğu gibi EKYK ve tepe-tırmanmalı EKYK yöntemleri en iyi ve benzer sonuçları çıkarmıştır. EYYK yöntemi en iyi başarımda iyileşme sağlamıştır. Daha sonra ERBYK gelmektedir. Tek başına RBYK yöntemi ise ERBYK yöntemine göre daha kötü bir sonuç çıkarmıştır. EBYK ise en iyi birey dahil edilmediğinde daha iyi bir iyileşme sağlamaktadır. Tırmanma sırasında başarımlar hesapları sayıldığında ise tepe-tırmanmalı EKYK yöntemleri yine büyük kromozom boyu nedeni ile çok iyi bir iyileşme sağlamamaktadırlar.



Şekil 6.5: 5 tepeli fonksiyon sonuçları



Şekil 6.6: 10 tepeli fonksiyon sonuçları

## 5. Sonuç ve Öneriler

Yerine koyma KDEA'lar için başarılı sonuçlar elde etmek açısından önem taşımaktadır. Hangi yerine koyma yönteminin kullanıldığına bağlı olarak çözülmek istenen problemde çözüme ulaşmak mümkün olabilir veya hiç mümkün olmayabilir. Bu çalışmada KDEA'larda yaygın bazı yerine koyma teknikleri ile tepe tırmanmanın da kullanıldığı yeni bir yöntem farklı yaklaşımlar altında deneysel olarak incelenmiştir.

Genel olarak EKYK yöntemi denenen tüm problemler için en iyi sonuçları çıkarmıştır. Lamarck yaklaşımı Darwin yaklaşımından daha etkili bir iyileşme sağlamaktadır. Tepe tırmanmalı yöntemler özellikle Lamarck yaklaşımı kullanıldığında, f5, f7 ve çok boyutlu çanta problemlerinde tırmanma başarımlar hesapları dahil edilmediğinde iyi sonuçlar çıkarsa da dahil edildiğinde iyi bir başarımlar iyileşmesi sağlamamaktadırlar. RBYK, EYYK ve EBYK yöntemlerinin ise problemden probleme farklılık göstermekle birlikte genel olarak en iyi bireyin dahil edilmediği durumlarda daha başarılı oldukları gözlemlenmiştir.

Çalışmada tepe tırmanmada en dik iniş yöntemi kullanılmıştır. Bu yöntemde bireyin tüm komşularına bakıldığından tırmanma süresi uzun olmaktadır. Bir sonraki aşamada diğer optimizasyon yöntemleri de kullanılabilir. Örneğin "greedy descent" yöntemi kullanıldığında bireyin komşularına rasgele bir düzende bakılıp, ilk bulunan iyi bireyden devam edildiğinden hesaplama zamanı daha az olabilir. Bu yöntem kullanıldığında hesaplama zamanı yanında toplumdaki bireylerin çeşitliliğinin nasıl etkilendiği de ileriki çalışmalarda incelenebilecek konular olabilir.

## 6. Kaynaklar

- [1] A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Algorithms*, Springer, 2003.
- [2] Jens Gottlieb, *Evolutionary Algorithms for Constrained Optimization Problems*. Ph.D. Thesis, Technical University of Clausthal, Clausthal, Germany, 1999.

- [4] T. Bäck, J.M. de Graaf, J.N. Kok, W.A. Kusters. Theory of genetic algorithms. *Bulletin of the EATCS*, 63:161-192, 1997.
- [5] Smith, J. E., Vavak, F.: Replacement Strategies in Steady State Genetic Algorithms: Static Environments, in: *Foundations of Genetic Algorithms 5* (W. Banzhaf, C. Reeves, Eds.), Morgan Kaufmann, San Francisco, CA, 1999, pp. 219-233.
- [6] J.K. van der Hauw. Evaluating and improving steady state evolutionary algorithms on constraint satisfaction problems. MSc thesis, Leiden Uni, 1996. (<http://www.liacs.nl/MScThesis/IR96-21.html>)
- [7] Digalakis, J.G., Margaritis, K. G., An experimental study of benchmarking functions for genetic algorithms, *Intern. J. Computer Math.*, 2002, 79(4), pp. 403\_416.
- [9] T. Bäck, "A User's guide to GENESYs 1.0," Dept. of Computer Science, University of Dortmund, 1992.
- [10] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, Chichester, West Sussex, England, 1990
- [11] J. E. Beasley, OR-Library, "Multidimensional knapsack problem"  
<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/mknapiinfo.htm>
- [12] "Multimodality Generator for Evolutionary Algorithms"  
<http://www.cs.uwyo.edu/~wspears/multi.html>
- [13] Eric W. Weisstein. "Method of Steepest Descent." From *MathWorld*, A Wolfram Web Resource. <http://mathworld.wolfram.com/MethodofSteepestDescent.html>