



**T.C.
BİLECİK ÜNİVERSİTESİ
BİLİMSEL ARAŞTIRMA PROJESİ
SONUÇ RAPORU**

**KENDİLİĞİNDEN UYARLANABİLİR KARŞITLIK
TABANLI FARKSAL GELİŞİM ALGORİTMASI**

PROJE YÖNETİCİSİ: YRD. DOÇ. DR. UĞUR YÜZGEÇ

ARAŞTIRMACILAR:

BAŞLAMA TARİHİ: 01 EKİM 2010

BİTİŞ TARİHİ: 01 EKİM 2011

**BİLECİK ÜNİVERSİTESİ
BİLECİK 2011**

ÖZET

Farksal Gelişim (Differential Algorithm, DE) çok amaçlı, lineer olmayan, türevlenemeyen fonksiyonları içeren problemlerin çözümü için kullanılan evrimsel optimizasyon (en iyileme) yöntemlerinden birisidir. DE Algoritmasının temel fikri popülasyondaki iki bireyin arasındaki farkın bir üçüncü bireye ilave edilmesidir. Genetik Algoritmalarından farklı olarak, DE algoritması tüm optimizasyon süreci boyunca sabit olan çaprazlama oranı, ölçekleme faktörü ve popülasyon boyu gibi birkaç kontrol parametresine sahiptir ve kendi yapısı içerisinde kayan noktalı sayıların gerçek kodlaması kullanılır. Kontrol parametrelerinin değerleri arama algoritmasının verimi ve elde edilen çözümün kalitesini artırmak için dikkatli bir şekilde seçilmelidir. DE algoritmasının verimliliği ve gürbüzlüğü tam olarak bu kontrol parametrelerinin ayarlanmasına bağlıdır. Diğer yandan, eğer DE algoritmasının kontrol parametreleri doğru şekilde seçilmez ise, DE beklenmedik şekilde zorlaşabilir. Karşıtlık Temelli Farksal Gelişim Algoritmasının fikri (Opposition based on Differential Evolution, ODE) temel olarak her bir jenerasyonda daha iyi bir çözüm bulmak için eş zamanlı bir tahmin ve onun karşıtı uygun bir tahmin göz önüne alınması esasına dayanır. Olasılık teorisine göre, eğer bir nokta çözümden uzak ise, o noktanın karşıtı o noktadan çözüme daha yakındır. Bu projede önerilen rastgele işlemlere dayanan yeni kendiliğinden uyarlanabilir yaklaşımlı karşıtlık temelli DE Algoritmasıyla ODE Algoritmasının kontrol parametreleri ayarlanır. Kontrol parametrelerinin daha iyi değerleri kısa bir sürede çözüme ulaşmayı sağlar. Bu projenin amacı optimizasyon sırasında farksal gelişim algoritmasının işlemsel yükünü azaltmak ve arama performansını artırmaktır. Önerilen algoritmanın performansını göstermek için bu projede literatürden farklı optimizasyon problemleri kullanılacaktır. Tüm farklı klasik, karşıtlık temelli, kendiliğinden uyarlanabilir ve uyarlanabilir karşıtlık temelli mekanizmalar için karşılaştırma çalışması yapılacaktır ve bu yeni yaklaşım değerlendirilecektir.

Anahtar Kelimeler

Genetik Algoritmalar, Farksal Gelişim Algoritması, Optimizasyon, Kendiliğinden Uyarlanabilir Yaklaşım

ABSTRACT

Differential evolution (DE) is one of the evolutionary optimization methods used for solving the problems which include nondifferentiable, nonlinear and multi-objective functions. The main idea of DE algorithm is that the difference value between two individuals adds to a third individual in population. Unlike the basic genetic algorithm (GA), DE has a few control parameters, such as scaling factor (SF), crossover probability constant (CR) and population size (PS), which are fixed during the whole optimization process, and the real coding of floating numbers is used in its own structure. The values of the control parameters should be selected carefully to increase the quality of the solution obtained and the efficiency of the search algorithm. The robustness and effectiveness of DE algorithm directly depend on the settings of these control parameters. On the other hand, DE can be unexpectedly complicated if the control parameters of DE are not properly selected. The idea of opposition based on differential evolution (ODE) is basically based on the concurrent consideration of an estimate and its corresponding opposite estimate to find a better solution in each generation. According to probability theory, if a point is far from the solution, the opposite of that point is closer to the solution than that point. By the opposition-based DE algorithm with new self adaptive approach, which is based on random procedures, proposed in this project, the control parameters of ODE are adjusted. The better values of the control parameters provide to find the solution in a short time. The aim of this project is to increase the search performance and to decrease the computational load of differential evolution algorithm during optimization. To demonstrate the performance of the proposed algorithm, different optimization problems from literature will be used in this project. For all the different classic, opposition-based, self-adaptive and adaptive opposition-based mechanisms, the comparison study will be done and this new approach will be evaluated.

Keywords

Genetic Algorithm, Differential Evolution Algorithm, Optimization, Self Adaptive Approach

İçindekiler

ÖZET.....	2
ABSTRACT.....	3
İçindekiler	4
Şekiller Dizini	5
Tablolar Dizini	6
1. Giriş.....	7
2. Farksal Gelişim Algoritması (DE)	7
2.1. Mutasyon Operatörü.....	10
2.2. Çaprazlama Operatörü	10
2.3. Seçim Operatörü.....	11
3. Karşıtlık Temelli Farksal Gelişim Algoritması (ODE).....	11
3.1. Karşıtlık Temelli Populasyon Başlatma.....	11
3.2. Karşıtlık Temelli Jenerasyon Atlama.....	12
4. Adaptif Farksal Gelişim Algoritması (ADE)	12
5. Adaptif Karşıtlık Temelli Farksal Gelişim Algoritması (AODE).....	13
6. Optimizasyon Problemleri	14
7. Ekmek Mayası Fermantasyon Süreci.....	16
8. Optimizasyon Probleminin Tanımı	19
Sonuçlar ve Tartışma.....	20
Kaynaklar	25

Şekiller Dizini

Şekil 1. Farksal Gelişim Algoritmasının işleyiş şeması

Şekil 2. Klasik Farksal Gelişim Algoritmasının Pseudo kodu

Şekil 3. Küresel Fonksiyon

Şekil 4. Rosenbrock Vadisi (Muz Fonksiyonu)

Şekil 5. Basamak Fonksiyonu

Şekil 6. Tilki Delikleri Fonksiyonu

Şekil 7. Endüstriyel bir Kesikli Fermantasyon Süreci

Şekil 8. DE Algoritmasının IC I başlangıç koşulu için optimizasyon benzetim sonuçları

Şekil 9. ADE Algoritmasının IC III başlangıç koşulu için optimizasyon benzetim sonuçları

Şekil 10. AODE Algoritması ile optimizasyon benzetim sonuçları (Glikoz değişimi)

Şekil 11. AODE Algoritması ile optimizasyon benzetim sonuçları (Etanol değişimi)

Şekil 12. AODE Algoritması ile optimizasyon benzetim sonuçları (Biokütle değişimi)

Şekil 13. AODE Algoritması ile optimizasyon benzetim sonuçları (Oksijen değişimi)

Şekil 14. AODE Algoritması ile optimizasyon benzetim sonuçları (Büyüme Hızı)

Şekil 15. AODE Algoritması ile optimizasyon benzetim sonuçları (Besleme Hızı)

Tablolar Dizini

Tablo 1. Optimizasyon Problemleri

Tablo 2. Yarı kesikli ekmek mayası fermantasyon sürecindeki parametre değerleri

Tablo 3. Dört farklı başlangıç koşulu için DE tabanlı metotların performans analizi
(ST_01: DE/rand/1/bin ve ST_02: DE/rand-to-best/1/bin)

1. Giriş

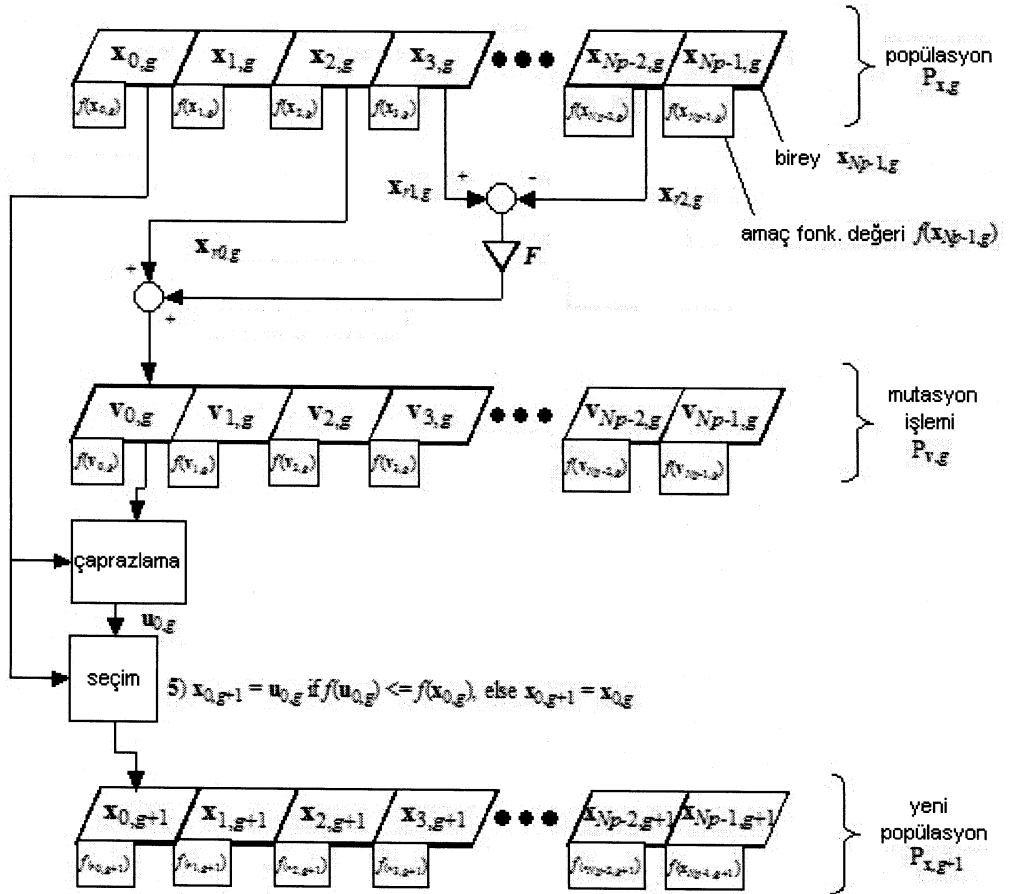
Optimizasyon problemleriyle karşılaşıldığında genellikle ilk yapılan, probleme özel sezgisel bir yaklaşım tekniği belirlemektir. Gelişime dayalı algoritmalar, genel metotlarla karşılaştırıldığında oldukça üstün performans göstermektedirler. Farksal Gelişim Algoritması basit, fakat güçlü popülasyon tabanlı bir algoritma olup, özellikle tamamen düzenlenmiş uzayda tanımlı ve gerçek değere sahip tasarım parametrelerini içeren fonksiyonların optimizasyonunda kullanılan bir direkt araştırma algoritmasıdır.

DE Algoritmasının temel fikri popülasyondaki iki bireyin arasındaki farkın bir üçüncü bireye ilave edilmesidir. Genetik Algoritmalarından farklı olarak, DE algoritması tüm optimizasyon süreci boyunca sabit olan çaprazlama oranı, ölçekleme faktörü ve popülasyon boyu gibi birkaç kontrol parametresine sahiptir ve kendi yapısı içerisinde kayan noktalı sayıların gerçek kodlamasını kullanır. Kontrol parametrelerinin değerleri arama algoritmasının verimi ve elde edilen çözümün kalitesini artırmak için dikkatli bir şekilde seçilmelidir. DE algoritmasının verimliliği ve gürbüzlüğü tam olarak bu kontrol parametrelerinin ayarlanmasına bağlıdır. Diğer yandan, eğer DE algoritmasının kontrol parametreleri doğru şekilde seçilmez ise, DE beklenmedik şekilde zorlaşabilir. Karşıtlık Temelli Farksal Gelişim Algoritmasının fikri (Opposition based on Differential Evolution, ODE) temel olarak her bir nesilde daha iyi bir çözüm bulmak için eş zamanlı bir tahmin ve onun karşıtı uygun bir tahmin göz önüne alınması esasına dayanır. Olasılık teorisine göre, eğer bir nokta çözümden uzak ise, o noktanın karşıtı o noktadan çözüme daha yakındır. Bu projede önerilen rastgele işlemlere dayanan yeni kendiliğinden uyarlanabilir karşıtlık temelli DE Algoritmasıyla ODE Algoritmasının kontrol parametrelerinin ayarlanması ve kontrol parametrelerinin ayarlanan bu değerleri ile çözüme daha kısa bir sürede ulaşması beklenmektedir. Bu projenin amacı optimizasyon sırasında farksal gelişim algoritmasının işlemsel yükünü azaltmak ve çözüm uzayındaki arama performansını artırmaktır.

2. Farksal Gelişim Algoritması (DE)

Hem fen bilimlerinde hem de sosyal bilimlerde ve bu bilim dallarının uygulama alanlarında karşılaşılan birçok problem, doğrusal veya doğrusal olmayan optimizasyon problemi olarak tanımlanmaktadır. Gerçek hayattaki uygulamalardaki problemlerin

büyük bir bölümü doğrusal olmayan bir yapıya sahiptir. Doğrusal olmayan problemlerin çözümüne yönelik olarak geliştirilmiş birçok teknik ve yaklaşım mevcuttur. Özellikle değişken sayısına ve veri tiplerine bağlı olarak problemlerin zorluk dereceleri de artabilmektedir. Bu tip problemlerin deterministik yöntemlerle çözümü, hem problemin yapısına bağlı olarak modellemede, hem de çözüm sürecinde zorluklar içermektedir. Bu yöntemlerde ise, ya istenilen sonuca ulaşılamamakta ya da kabul edilebilir sınırların dışındaki sürelerde sonuca ulaşılabilmektedir. Bunların üstesinden gelebilmek için sezgisel yöntemler geliştirilmiştir. Özellikle popülasyon temelli sezgisel çok noktalı arama işlemleri sayesinde, hızlı bir şekilde sonuç alınabilmektedir. Bu sezgisel yaklaşımlardan bazıları genetik algoritma (GA), ısı işlem, karınca kolonisi, benzetilmiş tavlama, bağışıklık ve farksal gelişim algoritmalarıdır. Farksal Gelişim Algoritması Price ve Storn tarafından 1995 yılında geliştirilmiş, özellikle sürekli verilerin söz konusu olduğu problemlerde etkin sonuçlar verebilen, isleyiş ve operatörleri itibariyle genetik almaya dayanan popülasyon temelli sezgisel optimizasyon tekniğidir [1-3]. Şekil 1'de DE algoritmasının temel işleyişi verilmiştir.



Şekil 1. Farksal Gelişim Algoritmasının işleyiş şeması

Temel olarak, DE algoritması populasyon içerisinde rastgele seçilen iki bireyin ağırlık farkının üçüncü bir bireye eklenmesi mantığına dayanmaktadır. Şekil 2'de DE algoritmasına ait pseudo kodu verilmektedir.

```

Input:  $Population_{size}, Problem_{size}, Weightingfactor, Crossover_{rate}$ 
Output:  $S_{best}$ 
Population  $\leftarrow$  InitializePopulation( $Population_{size}, Problem_{size}$ )
EvaluatePopulation(Population)
 $S_{best} \leftarrow$  GetBestSolution(Population)
While ( $\neg$  StopCondition())
  NewPopulation  $\leftarrow$   $\emptyset$ 
  For ( $P_i \in$  Population)
     $S_i \leftarrow$  NewSample( $P_i, Population, Problem_{size}, Weightingfactor, Crossover_{rate}$ )
    if ( $Cost(S_i) \leq Cost(P_i)$ )
      NewPopulation  $\leftarrow$   $S_i$ 
    else
      NewPopulation  $\leftarrow$   $P_i$ 
    end
  end
  Population  $\leftarrow$  NewPopulation
  EvaluatePopulation(Population)
   $S_{best} \leftarrow$  GetBestSolution(Population)
end
Return ( $S_{best}$ )

```

Şekil 2. Klasik Farksal Gelişim Algoritmasının Pseudo kodu

DE algoritmasının farklı çeşitleri literatürde kullanılmaktadır:

- en iyi birey / 1 fark vektörü / üstel çaprazlama
- rastgele bireyler / 1 fark vektörü / üstel çaprazlama
- rastgele ve en iyi bireyler / 1 fark vektörü / üstel çaprazlama
- en iyi birey / 2 fark vektörü / üstel çaprazlama
- rastgele bireyler / 2 fark vektörü / üstel çaprazlama
- en iyi birey / 1 fark vektörü / binom çaprazlama
- rastgele bireyler / 1 fark vektörü / binom çaprazlama
- rastgele ve en iyi bireyler / 1 fark vektörü / binom çaprazlama
- en iyi birey / 2 fark vektörü / binom çaprazlama
- rastgele bireyler / 2 fark vektörü / binom çaprazlama

Bu genel gösterim tarzında, ilk kısım populasyon içinden seçilen bireyleri, ikinci kısım seçilen bireylerin oluşturacağı fark vektörünün sayısını, son kısım ise çaprazlama tipini belirtmektedir. Bu proje kapsamında pratikte [3-4] en sık kullanılan iki stratejisi

rastgele ve en iyi bireyler / 1 fark vektörü / binom çaprazlama ile rastgele bireyler / 1 fark vektörü / binom çaprazlama olarak seçilmiştir. DE algoritması 3 önemli parametreye sahiptir: skala faktörü (SF), çaprazlama olasılık sabiti (CR) ve populasyon boyu (PS). Bir populasyon PS kadar bireyi içerir. Skala faktörü (SF) çaprazlama olasılık sabitine (CR) göre daha hassastır [1-2]. DE algoritmasının başında PS optimizasyon parametrelerine göre belirlenir ve optimizasyon sürecinde değiştirilmez. Normal şartlarda populasyon boyu (PS) optimizasyon parametre sayısının 5-10 katı olarak seçilir. Başlangıç populasyonu rastgele seçilir. DE algoritması 3 temel operatöre sahiptir: mutasyon, çaprazlama ve seçim [3]. Mutasyon ve çaprazlama operatörleri yeni bireyler üretir ve seçim operatörü ile uygun olanlar belirlenir, bu şekilde populasyonda sürekli en iyi bireylerin bulunması sağlanır.

2.1. Mutasyon Operatörü

Her iki DE algoritma stratejisi için mutasyona uğramış bireyler $v_{i,g}$ aşağıdaki şekilde bulunur:

$$v_{i,g+1} = x_{i,g} + SF(x_{b,g} - x_{i,g}) + SF(x_{r_1,g} - x_{r_2,g}) \quad (1)$$

$$v_{i,g+1} = x_{r_3,g} + SF(x_{r_1,g} - x_{r_2,g}) \quad (2)$$

Bu denklemlerde $r_1, r_2, r_3 \in [1, PS]$ rastgele katsayılardır. $x_{b,g}$ populasyon içerisindeki en iyi maliyet fonksiyonuna sahip bireyi göstermektedir. Rastgele katsayılar birbirinden farklı olmak zorundadır. Skala faktörü (SF) sabit bir değerdir ve $0 \leq SF \leq 2$ olarak değişir. Skala faktörü rastgele seçilen bireylerin fark vektörlerinin yükseltilmesinin kontrolünde kullanılır.

2.2. Çaprazlama Operatörü

Yeni bireyler çaprazlama operatörü ile aşağıdaki şekilde bulunabilir:

$$u_{i,g+1} = \begin{cases} v_{i,g+1}, & \text{if } r \leq CR \\ x_{i,g}, & \text{if } r > CR \end{cases} \quad (3)$$

Bu denklemde $r \in [0, 1]$ rastgele sayı üreticini ve $u_{i,g}$ deneme bireylerini göstermektedir. Çaprazlama olasılık sabiti (CR) $\in [0, 1]$ genellikle kullanıcı tarafından belirlenir. Eğer $CR = 1$ ise, oluşacak her yeni birey mutant olarak belirlenir $v_{i,g+1}$, yani rastgele seçilen fark vektörüne göre seçilir. Diğer taraftan $CR = 0$ durumunda, tüm

bireyler önceki jenerasyondan gelir $x_{i,g}$. Diğer bir deyişle, populasyonda değişim olmaz, yeni bireyler oluşmaz. Rastgele sayı üretici, mutant bireylerden $v_{i,g+1}$ en az birinden bir sonraki jenerasyona birey $u_{i,g+1}$ üretilmesini garanti eder.

2.3. Seçim Operatörü

Bir sonraki jenerasyondaki bireylerin $x_{i,g}$ tespitinde seçim operatörünün işletilmesi sırasında, deneme bireyinin $u_{i,g+1}$ maliyet fonksiyonu değeri hedef bireyin $x_{i,g}$ maliyet fonksiyonu ile karşılaştırılır ve sonraki jenerasyona optimizasyon problemine göre yeni birey seçilir. Aşağıdaki eşitlikte maksimum problemi için seçim operatörünün denklemi verilmiştir:

$$x_{i,g+1} = \begin{cases} u_{i,g+1}, & \text{if } f(u_{i,g+1}) < f(x_{i,g}) \\ x_{i,g}, & \text{otherwise} \end{cases} \quad (4)$$

Burada f maliyet fonksiyonunu göstermektedir. Eğer deneme bireyi $u_{i,g+1}$ hedef bireyden $x_{i,g}$ daha iyi bir maliyet fonksiyonuna sahipse, bir sonraki jenerasyonda kullanılacak birey $x_{i,g+1}$ deneme bireyi $u_{i,g+1}$ olarak seçilir. Aksi durumda populasyondaki hedef bireyde değişiklik olmaz.

3. Karşıtlık Temelli Farksal Gelişim Algoritması (ODE)

Karşıtlık temelli farksal gelişim (ODE) algoritması Tizhoosh (2005) and Rahnamayan et al. (2006) tarafından önerilen temel olarak eş zamanlı herbir jenerasyonda en iyi çözümü elde etmek için öngörü uzayı ile bu uzayın karşıtı diğer uzayı da kullanması fikrine dayanan bir algoritmadır. Olasılık teorisine göre, eğer bir nokta çözümden uzaksa, bu noktanın karşıtı çözüme daha yakındır. ODE iki temel adımdan oluşur: karşıtlık temelli populasyon başlatma ve karşıtlık temelli jenerasyon atlama.

3.1. Karşıtlık Temelli Populasyon Başlatma

Çözüm hakkında önceki bilgi olmadığı durumda, genel olarak başlangıçta populasyondaki bireyler rastgele seçilmektedir. Rastgele bireylerin seçilmesine ilave olarak, onların karşıt bireyleri karşıtlık temelli populasyon başlatma yardımıyla

hesaplanır. Böylece, ODE algoritması en uygun bireyler ile başlar. Bu işlemler aşağıda sırasıyla verilmektedir:

1. Rastgele popülasyondaki bireyler seçilir.
2. Aşağıdaki eşitlik yardımıyla başlangıç popülasyonundaki bireylerin karşıtları bulunur:

$$\tilde{x}_{i,1} = a + b - x_{i,1} \quad (5)$$

Burada $i = 1, 2, \dots, PS$, a ve b önceden tanımlanmış olan sınır değerlerini, $\tilde{x}_{i,1}$ ise ilk jenerasyondaki karşıt bireyi göstermektedir.

3. Bireylerin maliyet fonksiyonlarını hesaplamak için kullanılan fonksiyonun $f(\cdot)$ ile gösterildiğini varsayarsak, başlangıç popülasyonu için $f(\tilde{x}_{i,1})$ ile $f(x_{i,1})$ karşılaştırılarak, en uygun bireyler seçilir.

3.2. Karşıtlık Temelli Jenerasyon Atlama

Mutasyon, çaprazlama ve seçim operatörleri ile yeni jenerasyonun elde edilmesinden sonra, önceden tanımlanmış bir atlama olasılık değeri yardımıyla (J_r) karşıt popülasyon hesaplanır ve mevcut popülasyon ile karşıt popülasyon arasından popülasyon boyu (PS) kadar uygun birey seçilir. Karşıtlık temelli jenerasyon atlama işlemi atlama hızına (J_r) bağlıdır ve $[0,1]$ aralığında değişen rastgele bir sayıdır. Her bir jenerasyon sonunda rastgele üretilen $[0,1]$ aralığındaki sayı ile atlama hızı değeri karşılaştırılır, eğer $\text{rand}(0,1) < J_r$ ise, jenerasyon atlaması gerçekleştirilir. Jenerasyon atlamasında, önemli bir nokta önceden tanımlı sınır değerleri yerine mevcut popülasyondaki maksimum ve minimum değerlerin kullanılmasıdır. Klasik DE algoritmasının aksine, yeni bir kontrol parametresi olarak atlama hızı (J_r) DE parametrelerine eklenmiştir. Rahnamayan *et al.*(2006, 2008) çalışmalarına göre, optimal atlama hızı problem tabanlı bir değerdir. Yüksek boyutlu problemlerde küçük değer seçilmesi önerilmektedir.

4. Adaptif Farksal Gelişim Algoritması (ADE)

Brest *et al.* (2006) tarafından geliştirilen bu algorithmada skala faktörü (SF), çaprazlama olasılık sabiti (CR) gibi kontrol parametreleri rastgele işlemlere bağlı olarak kendiliğinden adaptif bir yaklaşımla ayarlanır. Kısa zamanda çözüme varmayı kontrol

parametrelerinin en iyi değerleri sağlamaktadır. Adaptif kontrol parametreleri $SF_{i,g+1}$ ve $CR_{i,g+1}$ aşağıdaki eşitlikler yardımıyla bulunabilir:

$$SF_{i,g+1} = \begin{cases} SF_{low} + rnd_1 * SF_{up}, & \text{if } rnd_2 \leq \tau_1 \\ SF_{i,g}, & \text{otherwise} \end{cases} \quad (6)$$

$$CR_{i,g+1} = \begin{cases} CR_{low} + rnd_3 * CR_{up}, & \text{if } rnd_4 \leq \tau_2 \\ CR_{i,g}, & \text{otherwise} \end{cases} \quad (7)$$

τ_1 ve τ_2 katsayıları SF ve CR değerlerini ayarlayan olasılık değerleridir. rnd_k rastgele uniform sayılardır ve $k \in \{1,2,3,4\}$. Brest *et al.* (2006) tarafından önerilen adaptif yapıdan farklı olarak çaprazlama olasılık sabiti (CR) hesabında rnd_3 ifadesi yerine $(CR_{low} + rnd_3 * CR_{up})$ ifadesi yerleştirilmiştir. Literatürde SF katsayısı nadiren birden büyük seçilmektedir. Kontrol parametresi $SF = 0$ olması durumunda, mutasyon işlemi çaprazlama işlemi gerçekleşmesine rağmen başarısızdır. $SF_{low} = 0.1$ ve $SF_{up} = 0.9$ olarak seçilmesi durumunda, yeni skala faktörü (SF) 0.1-1.0 arasında rastgele bir değer olarak düşünülür. Yeni CR değeri 0.75 ile 1.0 arasında değişmektedir. CR_{low} ve CR_{up} 0.75 ve 0.25 olarak düşünülmüştür. CR katsayısının en yüksek değerinde bireylerin çoğunluğu mutasyon vektöründen gelir. Böylece yeni bireylerin sayısı artar ve seçim işleminde yeni bireylerin şansı artar. $(g + 1)$ jenerasyonunda, kontrol parametreleri $SF_{i,g+1}$ ve $CR_{i,g+1}$ buradaki adaptif yapı ile mutasyon çaprazlama ve seçim operatörlerinden önce belirlenmektedir. Kontrol parametrelerinin sınır değerleri sabit değerlerdir. Benzer şekilde τ_1 ve τ_2 olasılık değerleri sabit olarak seçilmektedir. Denemeler ışığında bu olasılık değerleri 0.5 olarak seçilmiştir. Bu şekilde, optimizasyon probleminin çözümü daha kısa sürede ve daha hızlı olarak elde edilir.

5. Adaptif Karşıtlık Temelli Farksal Gelişim Algoritması (AODE)

Bu projede önerilen yeni algoritma ODE yapısı ile adaptif yaklaşımın birleşimine dayanmaktadır. Bu önerilen algorithmada, ADE algoritmasındaki adaptif parametrelere, ODE algoritmasının kontrol parametresi (J_r) atlama hızı aşağıdaki verilen eşitliğe bağlı olarak hesaplanmaktadır:

$$J_{r,g+1} = \begin{cases} J_{r,low} + rnd_5 * J_{r,up}, & \text{if } rnd_6 \leq \tau_3 \\ J_{r,g}, & \text{otherwise} \end{cases} \quad (8)$$

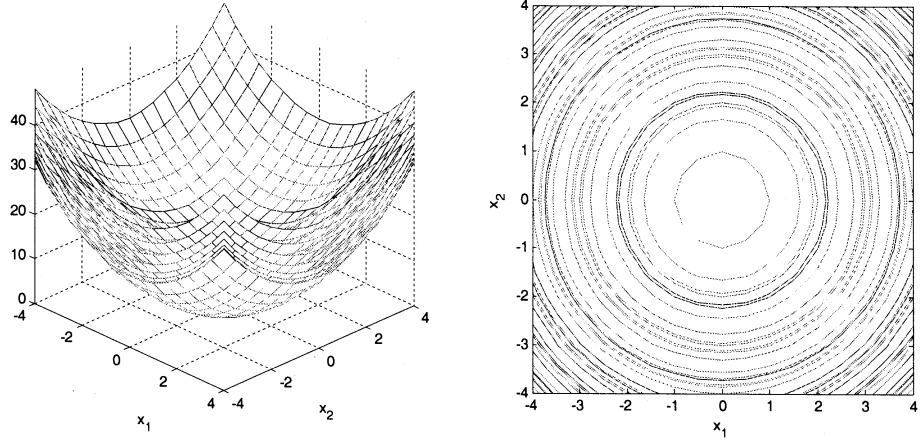
Buradaki eşitlikte rnd_5 ve rnd_6 [0,1] aralığında değişen rastgele sayıları, τ_3 atlama hızının (J_r) ayarlanması için olasılık değerini göstermektedir. $J_{r,low}$ ve $J_{r,up}$ atlama hızının sınır değerlerini temsil etmektedir. Bu değerler 0.05 ve 0.25 olarak seçilmiştir. Sonuç olarak yeni atlama hızı (J_r) 0.05-0.3 arasında değişir. ADE algoritmasındaki kontrol parametrelerinin (skala faktörü (SF), çaprazlama olasılık sabiti (CR)) ayarlanmasından farklı olarak atlama hızı her jenerasyonda yalnızca bir kere ayarlanmaz, jenerasyonda her birey için ayarlanmaktadır.

6. Optimizasyon Problemleri

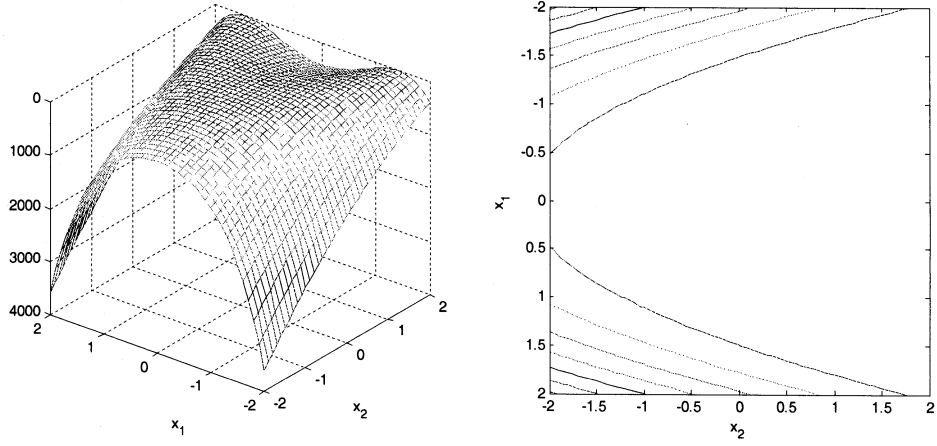
Matematiksel temelli veya istatistik yöntemlere dayalı veya rastgele temellere dayanan algoritmaların test edilmesi için literatürde yer alan birçok test problemine rastlamak mümkündür. Tablo 1'de literatürde kullanılan bazı optimizasyon test problemleri verilmiştir. Şekil 3-Şekil 6 sırasıyla bu fonksiyonların değişimlerini üç boyutlu ve iki boyutlu düzlemlerde göstermektedir. Bu fonksiyonların çizimlerinde diğer değişkenler sabit seçilerek grafikler elde edilmiştir. Literatürdeki genel optimizasyon problemlerinin başında küresel fonksiyon gelmektedir. Rosenbrock vadisi zor bir optimizasyon problemi olarak yer almaktadır ve genel çözüm algoritmaları düzlük alanda takılmaktadır. Basamak ve Tilki delikleri fonksiyonlarının da global çözümünün bulunmasında klasik algoritmalar lokal (yerel) minimum noktalarına takılabilmektedir.

Tablo 1. Optimizasyon Problemleri

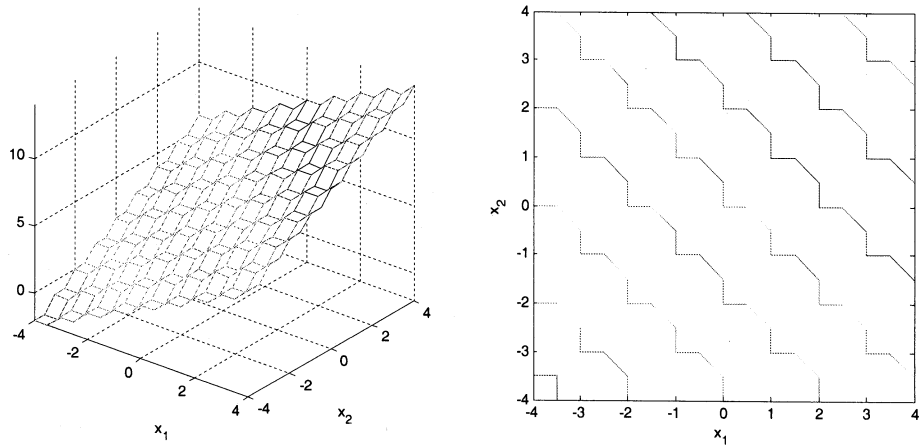
Küresel Fonksiyon	$f(x) = \sum_{i=1}^3 x_i^2$	(1)
Rosenbrock Vadisi (Muz Fonksiyonu)	$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	(2)
Basamak Fonksiyonu	$f(x) = \sum_{i=1}^5 \ x_i\ $	(3)
Tilki Delikleri Fonksiyonu	$f(x) = \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}$ $a_{ij} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{bmatrix}$	(4)



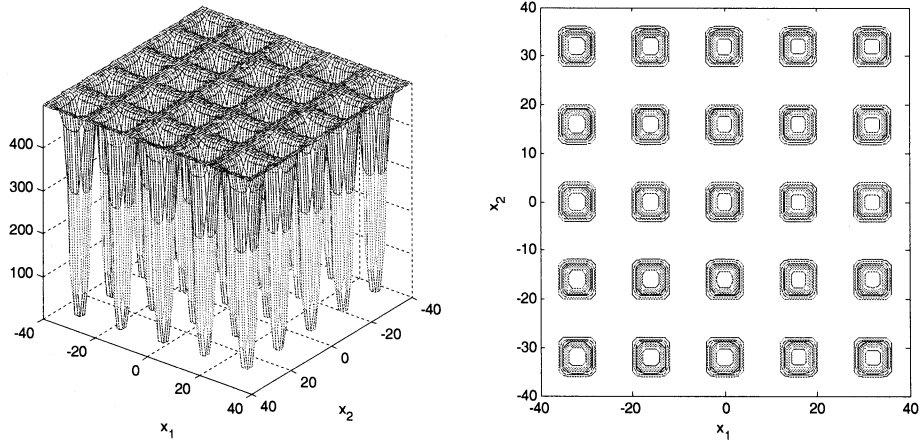
Şekil 3. Küresel Fonksiyon



Şekil 4. Rosenbrock Vadisi (Muz Fonksiyonu)



Şekil 5. Basamak Fonksiyonu

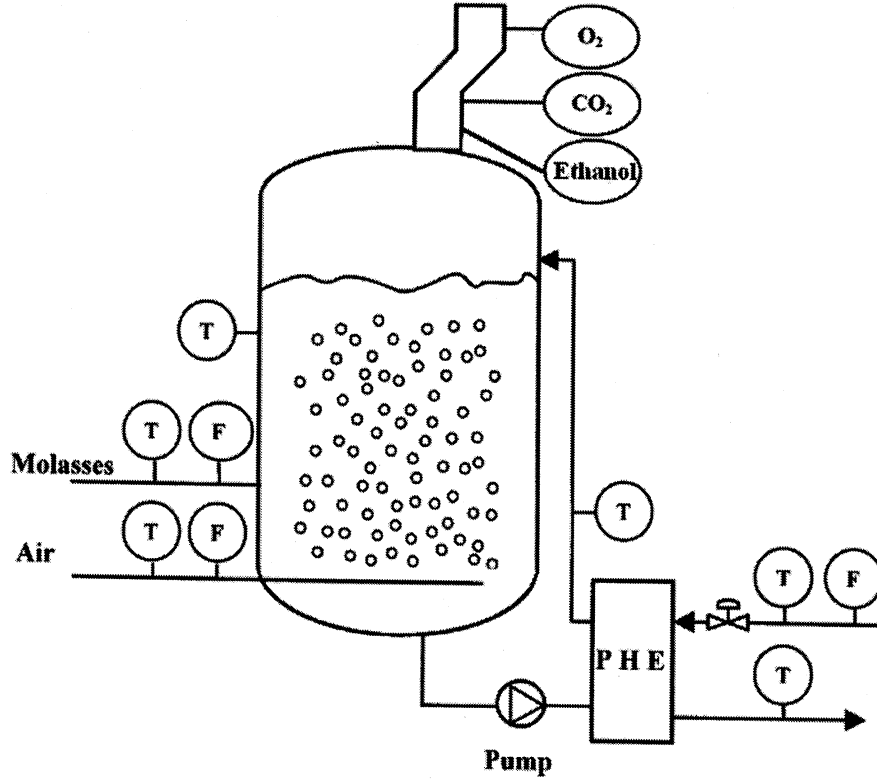


Şekil 6. Tilki Delikleri Fonksiyonu

7. Ekmek Mayası Fermantasyon Süreci

Ekmek mayası fermantasyon sürecinde, optimal çalışma koşulları ürün kalitesini sağlamak ve verimi arttırmak için bulunması gerekli noktadır. Sonnleitner and Kappeli [14]'e göre ekmek mayası fermantasyon sürecinde üç farklı metabolik yol vardır: glikoz oksidasyonu, glikoz azalımı ve etanol oksidasyonu. Bu metabolik yollar ekmek mayası hücrelerinin kapasitesi ve besleme akışına bağlıdır. Şekil 7'de kesikli fermantasyon sürecinin genel hatları ana bileşenleri ile gösterilmiştir. Bu süreçte harici bir ısı kaynağı (PHE), sıcaklık ölçümleri alabilmek için sıcaklık ölçüm noktaları (T), akış hızlarını ölçmek için akış metre (F) ve çıkış bacasında O₂, CO₂ ve etanol ölçümleri için gaz analizörü kullanılmaktadır.

Eğer glikoz konsantrasyonu aşırı bir şekilde fermentörde olursa, en son ürün konsantrasyonu yüksek olacaktır, fakat aynı zamanda ürün verimi aşırı etanol oluşumu nedeniyle düşük olacaktır. Ters durumda ise eğer yeterli miktarda glikoz fermentörde bulunmaz ise, üretim düşecek, fakat ürün verimi yüksek olacaktır. Bu üretim ve ürün verimi arasındaki karmaşayı yenebilmek için akışkan yatağa verilen beslemenin doğru olarak kontrol edilmesi gerekmektedir. Bu amaçla kullanılan akışkan yatak fermantasyon sürecinin modeli simülasyon verilerini elde etmek için kullanılmıştır. Fermantasyon modeli beşinci dereceden nonlinear matematiksel bir modeldir ve 12 diferansiyel denklem ve 5 aritmetik denklemden oluşur.



Şekil 7. Endüstriyel bir Kesikli Fermantasyon Süreci

Kinetik model Sonnleitner & Kappeli [14] tarafından geliştirilmiş olup, mayanın sınırlı oksijen kapasitesi olduğu, aşırı glikoz konsantrasyonu ve/veya oksijen sınırlaması koşulu altında etanol oluşumu olduğu varsayımına dayanmaktadır [15]. Yine model Karakuzu ve diğ. [16] tarafından oksidasyon kapasitesi ve glikoz tüketim hızı formülleri geliştirilmiştir. Kinetik model denklemleri aşağıda verilmiştir:

$$\text{Glikoz tüketim hızı:} \quad Q_s = Q_{s,\max} \frac{C_s}{K_s + C_s} (1 - e^{-t/t_d}) \quad (9)$$

$$\text{Oksidasyon kapasitesi:} \quad Q_{o,\text{lim}} = Q_{o,\max} \frac{C_o}{K_o + C_o} \frac{K_i}{K_i + C_e} \quad (10)$$

$$\text{Özel büyüme hızı limiti:} \quad Q_{s,\text{lim}} = \frac{\mu_{cr}}{Y_{x/s}^{ox}} \quad (11)$$

$$\text{Oksidatif glikoz metabolizması:} \quad Q_{s,ox} = \min \left(\begin{array}{c} Q_s \\ Q_{s,\text{lim}} \\ Q_{o,\text{lim}}/Y_{o/s} \end{array} \right) \quad (12)$$

$$\text{Redüktif glikoz metabolizması:} \quad Q_{s,red} = Q_s - Q_{s,ox} \quad (13)$$

$$\text{Etanol tüketim hızı: } Q_{e,up} = Q_{e,max} \left(\frac{C_e}{K_e + C_e} \right) \left(\frac{K_i}{K_i + C_e} \right) \quad (14)$$

$$\text{Oksidatif etanol metabolizması: } Q_{e,ox} = \min \left(\frac{Q_{e,up}}{(Q_{o,lim} - Q_{s,ox} Y_{o/s}) Y_{e/o}} \right) \quad (15)$$

$$\text{Etanol üretim hızı: } Q_{e,pr} = Q_{s,red} Y_{e/s} \quad (16)$$

$$\text{Toplam özel büyüme hızı: } \mu = Q_{s,ox} Y_{x/s}^{ox} + Q_{s,red} Y_{x/s}^{red} + Q_{e,ox} Y_{x/e} \quad (17)$$

$$\text{Karbondiyoksit üreme hızı: } Q_c = Q_{s,ox} Y_{c/s}^{ox} + Q_{s,red} Y_{c/s}^{red} + Q_{e,ox} Y_{c/e} \quad (18)$$

$$\text{Oksijen tüketim hızı: } Q_o = Q_{s,ox} Y_{o/s} + Q_{e,ox} Y_{o/e} \quad (19)$$

$$\text{Solunum katsayısı: } RQ = Q_c / Q_o \quad (20)$$

Dinamik model 5. dereceden lineer olmayan bir modeldir ve glikoz, etanol, oksijen ve biokütle konsantrasyonu ile tanımlanmış kütle balans eşitliklerine dayanmaktadır.

Model denklemleri aşağıda verilmiştir:

$$\frac{dC_s}{dt} = \frac{F}{V} (S_o - C_s) - \left(\frac{\mu}{Y_{x/s}^{ox}} + \frac{Q_{e,pr}}{Y_{e/s}} + Q_m \right) C_x \quad (21)$$

$$\frac{dC_o}{dt} = -Q_o C_x + k_L a_o (C_o^* - C_o) - \frac{F}{V} C_o \quad (22)$$

$$\frac{dC_e}{dt} = (Q_{e,pr} - Q_{e,ox}) C_x - \frac{F}{V} C_e \quad (23)$$

$$\frac{dC_x}{dt} = \mu C_x - \frac{F}{V} C_x \quad (24)$$

$$\frac{dV}{dt} = F \quad (25)$$

Tablo 2. Yarı kesikli ekmek mayası fermantasyon sürecindeki parametre değerleri

Parameter	Units	Value	Parameter	Units	Value	Parameter	Units	Value
K_e	g L^{-1}	0.1	$Y_{e/o}$	g g^{-1}	1.1236	$Q_{o,max}$	$\text{g g}^{-1}\text{h}^{-1}$	0.255
K_o	g L^{-1}	9.6×10^{-5}	$Y_{e/s}$	g g^{-1}	0.4859	$Q_{s,max}$	$\text{g g}^{-1}\text{h}^{-1}$	2.943
K_i	g L^{-1}	3.5	$Y_{x/e}$	g g^{-1}	0.7187	Q_m	$\text{g g}^{-1}\text{h}^{-1}$	0.03
K_s	g L^{-1}	0.612	$Y_{c/s}^{ox}$	g g^{-1}	0.5744	μ_{cr}	h^{-1}	0.21
$Y_{x/s}^{ox}$	g g^{-1}	0.585	$Y_{c/s}^{red}$	g g^{-1}	0.462	S_o	g L^{-1}	325
$Y_{x/s}^{red}$	g g^{-1}	0.05	$Y_{c/e}$	g g^{-1}	0.645	C_o^*	g L^{-1}	0.006
$Y_{o/s}$	g g^{-1}	0.3857	$Q_{e,max}$	$\text{g g}^{-1}\text{h}^{-1}$	0.238	$k_L a_o$	h^{-1}	700

Tablo 2'de literatürden [14,16] ve deneysel verilerden elde edilen süreç parametreleri verilmiştir.

8. Optimizasyon Probleminin Tanımı

Genel olarak fermantasyon sürecinde, ana hedef süreç sonundaki ürün miktarını maksimum da tutarak, süreç sırasındaki etanol oluşumunu minimize etmektir. Bu yüzden amaç fonksiyonunda yalnızca biokütle konsantrasyonu bulunmasından bahsetmek mümkün değildir. Başlangıç ve çalışma koşullarına bağlı olarak fermantasyon sırasında istenilmeyen bir durum olarak etanol konsantrasyonunda artışlar görülebilir. Eğer etanol oluşumu fermantasyon süreci boyunca aşırı miktarda artış gösteriyor ise, bu durum süreç sonunda oluşan ürün kalitesi ve değerinde bozulmalara yol açacaktır. Tersi durumda, besleme hızının azaltılması ile etanol oluşumu ve biokütle oluşumunun hızı yavaşlayacaktır. Maksimum son ürün elde etmek için, büyüme hızının kritik büyüme hızını (μ_{cr}) çok yakın takip etmesi gerekmektedir. Bu kritik değeri aştığında, etanol oluşumu tekrar artacaktır. Optimizasyon problemi üç kısım olarak düşünülmüştür: biokütle konsantrasyonunun maksimum olması, etanol oluşumunun minimumda tutulması ve ortalama büyüme hızının kritik büyüme hızına yakın elde etmektir. Sonuç olarak, optimizasyon problemi aşağıdaki şekilde verilmiştir:

$$\max_F J = J_1 - J_2 - J_3 \quad (26)$$

$$\text{subject to } 0 \leq t \leq t_f, F_{\min} \leq F \leq F_{\max} \quad (27)$$

$$J_1(t_i) = \beta_1 \Delta C_x(t_i) \quad (28)$$

$$J_2(t_i) = \beta_2 \int_{t_{i-1}}^{t_i} C_e(t) dt \quad (29)$$

$$J_3(t_i) = \beta_3 \left[\frac{1}{(t_i - t_{i-1})} \int_{t_{i-1}}^{t_i} \mu(t) dt - \mu_{cr} \right]^2 \quad (30)$$

$\beta_1, \beta_2, \beta_3$ optimizasyon ağırlık faktörleri ve t_i i. iterasyondaki zamanı göstermektedir. Kontrol değişkeni besleme hızı (F) olarak seçilmiştir ve 0 Lh^{-1} ile 3500 Lh^{-1} aralığında değişmektedir. Buna ilave olarak, sürecin çalışma koşulları optimizasyon işlemi ile sınırlıdır. Toplam biokütle hacmi fermentör hacminden küçük ($V(t) < V_{fer}$) olmak

zorundadır. Toplam süreç zamanı optimizasyon işlemi için hesaplama adımlarına bölünmüştür. Her bir zaman aralığında optimal çözüm araştırılır ve sürecin son zamana kadar bu işlem devam ettirilir.

Sonuçlar ve Tartışma

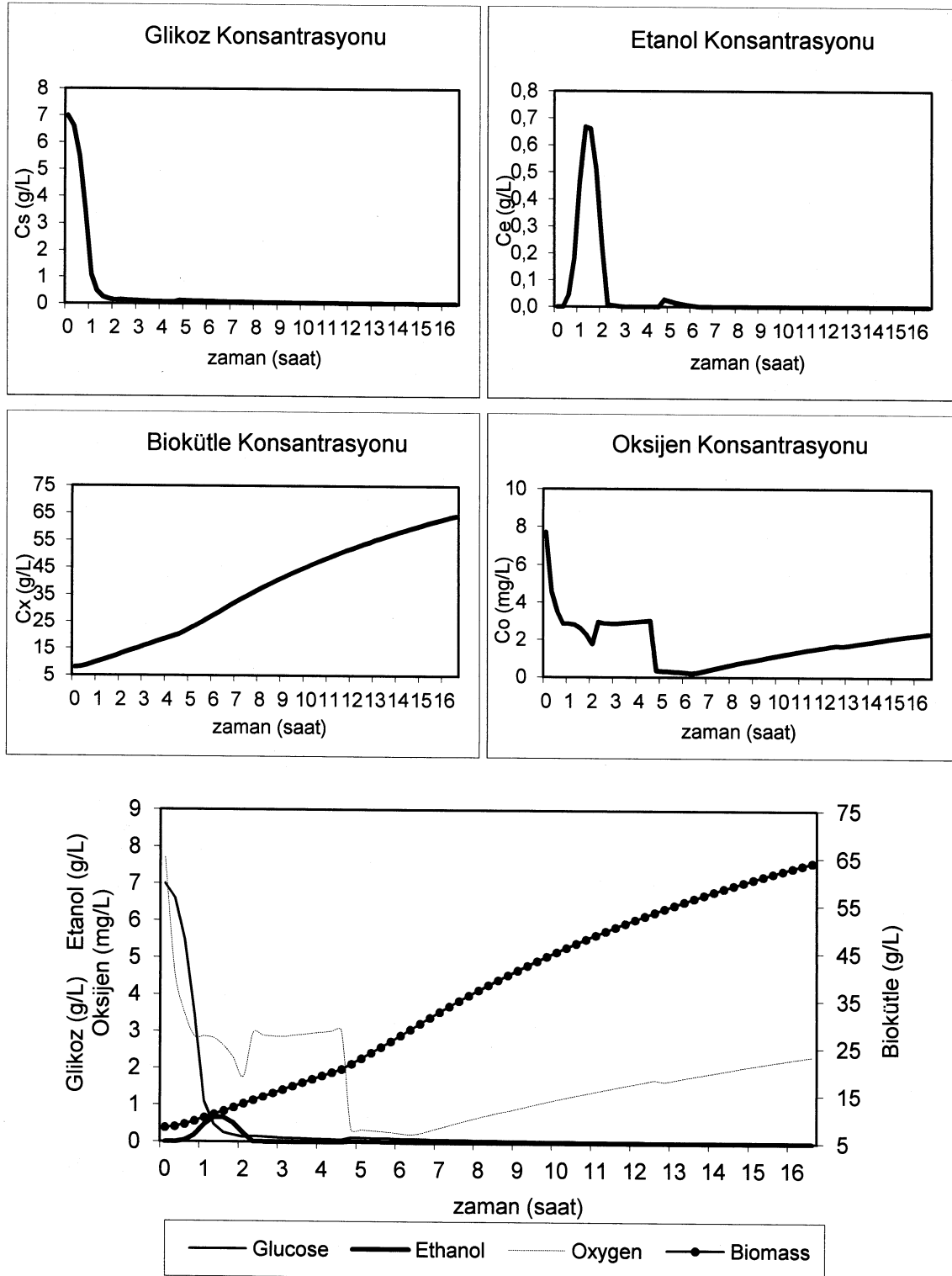
Önerilen Farksal Gelişim Algoritmalarını değerlendirmek için dört farklı başlangıç koşulu ile fermentasyon süreci modeli koşturulmuş ve bu algoritmaların performansları değerlendirilmiştir. Benzetimlerde Intel I7 860 CPU 2.8GHz, 8GB DDR3 RAM özelliklere sahip bir bilgisayar kullanılmıştır. Tüm program kodları Matlab platformunda yazılmıştır. Algoritmaların performans değerlendirmesi için kullanılan dört farklı başlangıç koşulu aşağıdaki gibi seçilmiştir:

- IC I: başlangıç biokütle ve glikoz konsantrasyonu yüksek.
($C_x(0) = 8 \text{ gL}^{-1}$, $C_s(0) = 7 \text{ gL}^{-1}$)
- IC II: başlangıç biokütle konsantrasyonu yüksek ve glikoz konsantrasyonu düşük.
($C_x(0) = 8 \text{ gL}^{-1}$, $C_s(0) = 0.1 \text{ gL}^{-1}$)
- IC III: başlangıç biokütle konsantrasyonu düşük ve glikoz konsantrasyonu yüksek.
($C_x(0) = 1.5 \text{ gL}^{-1}$, $C_s(0) = 7 \text{ gL}^{-1}$)
- IC IV: başlangıç biokütle ve glikoz konsantrasyonu düşük.
($C_x(0) = 0.5 \text{ gL}^{-1}$, $C_s(0) = 0.1 \text{ gL}^{-1}$)

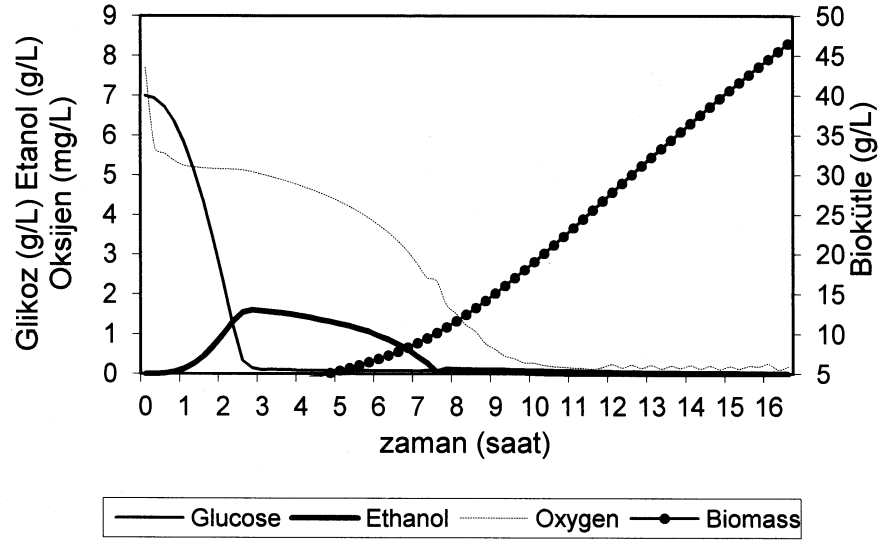
Çalışmada kullanılan DE algoritmalarının parametre değerleri aşağıda verilmiştir:

- Tüm algoritmalar için popülasyon boyu: $PS = 10$. $D = 10$ [3], [9].
- DE ve ODE algoritmaları için skala faktörü (SF): $SF = 0.5$ [9], [11].
- DE ve ODE algoritmaları için çaprazlama olasılık sabiti (CR): $CR = 0.9$ [9], [11].
- ODE algoritması için atlama hızı: $J_r = 0.3$ [11].
- Tüm algoritmalar için mutasyon ve çaprazlama stratejileri: DE/rand/1/bin (*Strategy I*) ve DE/rand-to-best/1/bin (*Strategy II*) [1], [12].
- ADE ve AODE algoritmaları için skala faktörünün sınır değerleri: $SF_{low} = 0.1$ ve $SF_{up} = 0.9$ [9].
- ADE ve AODE algoritmaları için çaprazlama olasılığının sınır değerleri: $CR_{low} = 0.75$ ve $CR_{up} = 0.25$.
- AODE algoritması için atlama hızının sınır değerleri: $J_{r,low} = 0.05$ ve $J_{r,up} = 0.25$.

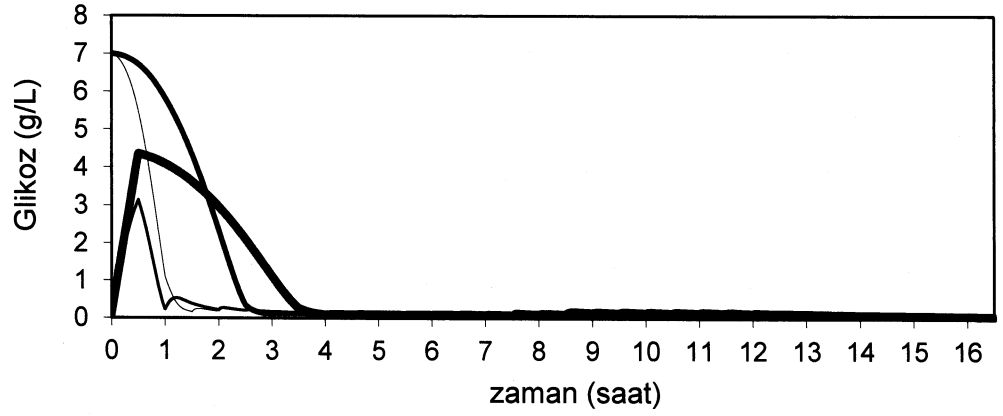
Şekil 8'de DE algoritması ile IC 1 başlangıç koşulu için fermantasyon sürecinin optimizasyon sonuçları gösterilmiştir. Şekil 9'da ise ADE algoritması ile yapılan optimizasyon sonuçları verilmiştir. Şekil 10-15'de dört farklı başlangıç koşulu için önerilen AODE algoritması ile yapılan optimizasyon sonuçları gösterilmiştir.



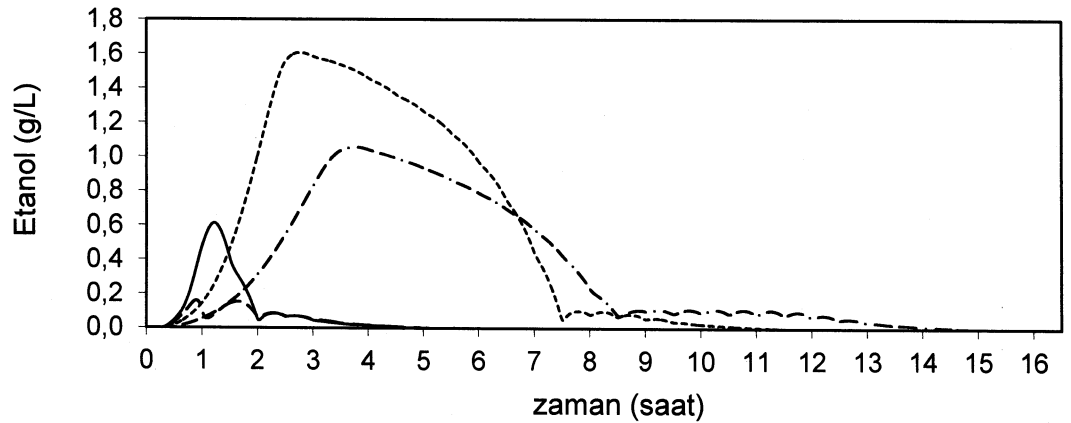
Şekil 8. DE Algoritmasının IC I başlangıç koşulu için optimizasyon benzetim sonuçları



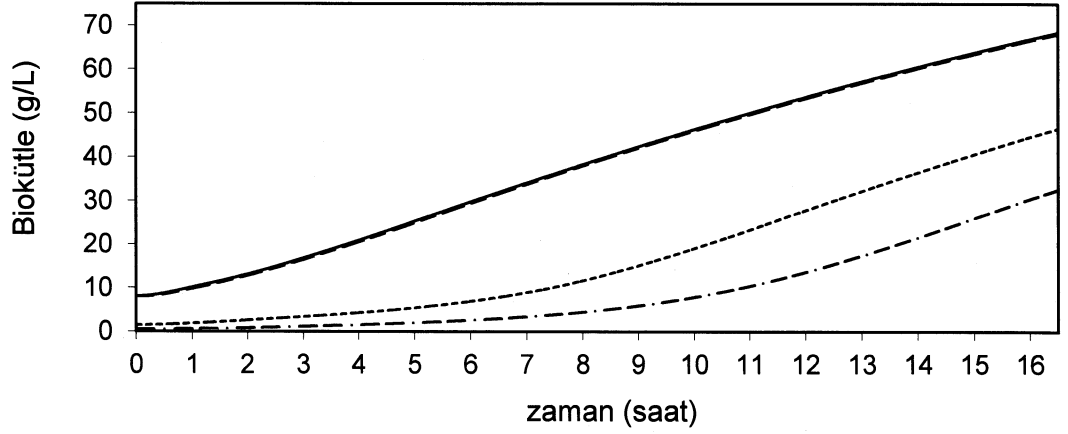
Şekil 9. ADE Algoritmasının IC III başlangıç koşulu için optimizasyon benzetim sonuçları



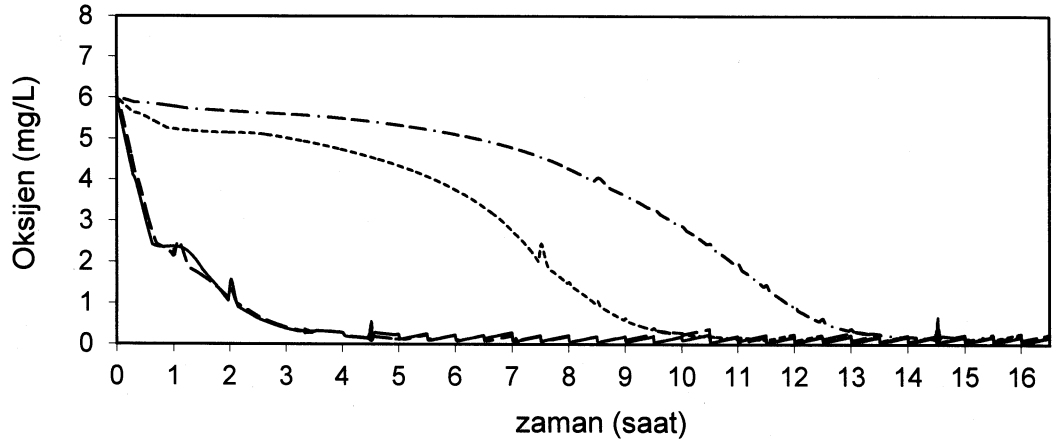
Şekil 10. AODE Algoritması ile optimizasyon benzetim sonuçları (Glukoz değişimi)



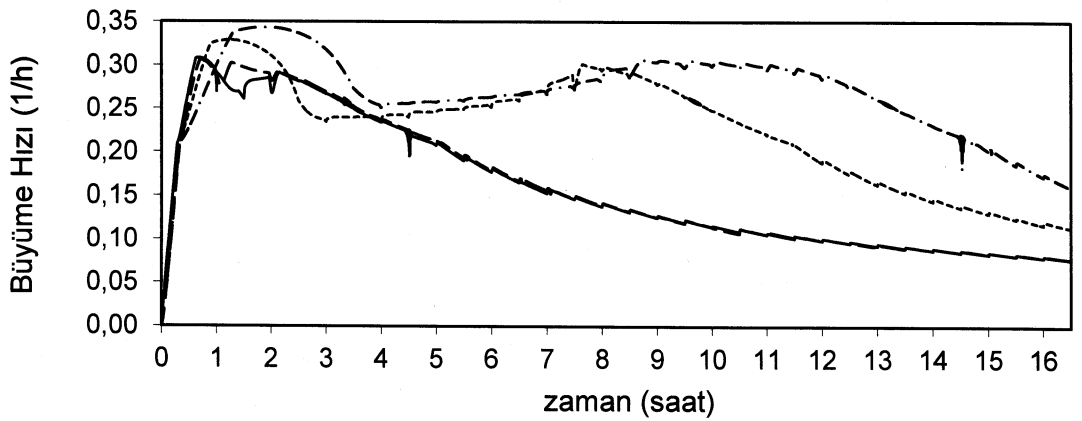
Şekil 11. AODE Algoritması ile optimizasyon benzetim sonuçları (Etanol değişimi)



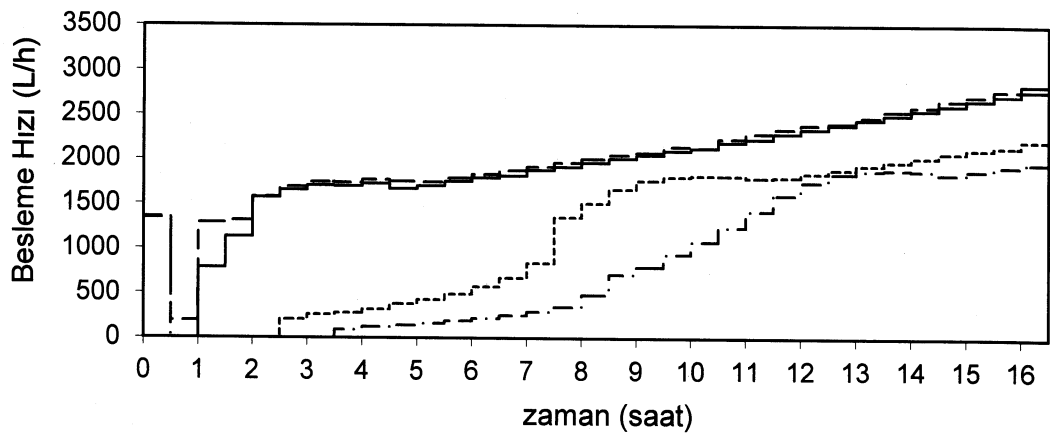
Şekil 12. AODE Algoritması ile optimizasyon benzetim sonuçları (Biokütle değişimi)



Şekil 13. AODE Algoritması ile optimizasyon benzetim sonuçları (Oksijen değişimi)



Şekil 14. AODE Algoritması ile optimizasyon benzetim sonuçları (Büyüme Hızı)



Şekil 15. AODE Algoritması ile optimizasyon benzetim sonuçları (Besleme Hızı)

Tablo 3'de dört farklı başlangıç durumu fermantasyon sürecinin optimizasyon probleminin DE, ADE, ODE ve önerilen AODE metotları ile çözüm sonuçları gösterilmiştir. Bu tabloda iki farklı temel DE stratejisi kullanılmıştır: DE/rand/1/bin ve DE/rand-to-best/1/bin.

Tablo 3. Dört farklı başlangıç koşulu için DE tabanlı metotların performans analizi (ST_01: DE/rand/1/bin ve ST_02: DE/rand-to-best/1/bin)

ST_01	DE			ADE			ODE			AODE		
	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}
IC I	67,039	0,820	0,159	68,206	0,601	0,159	67,406	0,942	0,161	68,34	0,604	0,159
IC II	60,882	0,687	0,155	68,124	0,264	0,160	67,330	0,561	0,162	68,039	0,273	0,160
IC III	45,848	7,334	0,226	46,302	7,000	0,227	46,366	6,987	0,227	46,382	7,047	0,227
IC IV	30,994	7,695	0,264	31,548	7,306	0,265	32,581	5,357	0,268	32,607	5,274	0,268

ST_02	DE			ADE			ODE			AODE		
	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}
IC I	67,985	0,603	0,159	68,249	0,592	0,159	67,423	0,941	0,161	68,421	0,605	0,159
IC II	67,883	0,277	0,160	68,041	0,270	0,160	67,090	0,606	0,162	68,146	0,274	0,160
IC III	45,930	8,132	0,226	46,553	6,869	0,227	44,749	7,943	0,227	46,521	7,100	0,222
IC IV	32,451	5,281	0,267	32,577	5,326	0,268	32,692	5,194	0,268	32,643	5,212	0,268

	GA			ADE			ODE			AODE		
	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}	C_x	C_e	μ_{ort}
IC I	65,514	1,194	0,161	68,249	0,592	0,159	67,406	0,942	0,161	68,421	0,605	0,159
IC II	64,928	0,753	0,161	68,041	0,270	0,160	67,330	0,561	0,162	68,146	0,274	0,160
IC III	43,877	5,655	0,223	46,553	6,869	0,227	46,366	6,987	0,227	46,521	7,100	0,222
IC IV	29,394	2,356	0,260	32,577	5,326	0,268	32,581	5,357	0,268	32,643	5,212	0,268

Benzetim sonuçlarından ve Tablo 3'deki performans tablosundan da görüleceği gibi önerilen AODE algoritması örnek olarak ele alınan fermantasyon süreci optimizasyon problemini başarı ile çözmüştür. Her iki strateji sonuçlarına bakıldığında ise ST_01'de IC II başlangıç koşulu için ADE algoritması en iyi çözümü vermiştir, fakat önerilen metot ile aralarında çok önemli bir fark bulunmamaktadır. Yine ST_02 için IC III başlangıç koşulunda AODE ile ADE algoritmalarının performansları birbirine oldukça yakındır. Önerilen AODE algoritması fermantasyon süreci gibi karmaşık çok değişkenli optimizasyon problemlerinde rahatlıkla kullanılabilir denilebilir.

Kaynaklar

- [1] Price, K., & Storn, R. (1997). Differential evolution: A simple evolution strategy for fast optimization. *Dr. Dobbs's J. Software Tools*, 22 (4), 18-24.
- [2] Storn, R., & Price, K. (1997). Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optimization*, 11, 341-359.
- [3] Price, K. (1999). An Introduction to Differential Evolution, D. Corne, M. Dorigo, and F. Glover, Eds. London, U.K.: McGraw-Hill, 1999, pp. 79-108, ISBN: 007-709506-5, *New Ideas in Optimization*.
- [4] Liu, J., & Lampinen, J. (2002). On setting the control parameter of the differential evolution method. In *Proc. 8th Int. conf. Soft Computing (MENDEL 2002)*, 11-18.
- [5] Liu, J., & Lampinen, J. (2005). A fuzzy adaptive differential evolution algorithm. *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, 9 (6), 448-462.
- [6] Ali, M. M., & Törn, A. (2004). Population set-based global optimization algorithms: Some modifications and numerical studies. *Comput. Oper. Res.*, 31(10), 1703-1725.
- [7] Babu, B. V., & Jehan, M. M. L. (2006). Modified differential evolution (MDE) for optimization of non-linear chemical process. *Computers & Chemical Eng.*, 30, 989-1002.
- [8] Sun, J., Zhang, Q., & Tsang, E. (2004). DE/EDA: A new evolutionary algorithm for global optimization. *Inf. Sci.*, 169, 249-262.
- [9] Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.*, 10(6), 646-657.
- [10] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2006). Opposition-based differential evolution algorithms. In *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2010-2017.
- [11] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2008). Opposition-based differential evolution. *IEEE Trans. Evol. Comput.*, 12(1), 64-79.

- [12] Babu, B. V., & Jehan, M. M. L. (2003). Differential evolution for multi-objective optimization. In Proceedings of 2003 Congress on Evolutionary Computation (CEC-2003), Canberra, Australia, 2696-2703.
- [13] Tizhoosh, H. R. (2005). Opposition-based learning: A new scheme for machine intelligence. in Proc. Int. Conf. Comput. Intell. Modeling Control and Autom., Vienna, Austria, 695-701.
- [14] Sonnleitner, B., & Kappeli, O. (1986). Growth of *saccharomyces cerevisiae* is controlled by its limited respiratory capacity: formulation and verification of a hypothesis. *Biotech. & Bioeng.*, 28, 927-937.
- [15] Berber, R., Pertev, C., & Türker, M. (1999). Optimization of feeding profile for baker's yeast production by dynamic programming. *Bioprocess Eng.*, 20, 263-269.
- [16] Karakuzu, C., Türker, M., & Öztürk, S. (2006). Modeling, on-line state estimation and fuzzy control of production scale fed-batch baker's yeast fermentation. *Control Eng. Practice*, 14, 959-974.