Konya Mühendislik Bilimleri Dergisi, c. 9, s. 1, 164-179, 2021
Konya Journal of Engineering Sciences, v. 9, n. 1, 164-179, 2021
ISSN: 2667-8055 (Elektronik)
DOI: 10.36306/konjes.738684

ENDÜSTRİYEL BİYOLOJİK FERMANTASYON İŞLEMİ İÇİN DENGE OPTİMİZASYON ALGORİTMASIYLA KONTROLÖR TASARIMI

Abdullah ATEŞ

İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya, TÜRKİYE
abdullah.ates@inonu.edu.tr

(Geliş/Received: 17.05.2020; Kabul/Accepted in Revised Form: 21.11.2020)

ÖZ: Bu çalışmada yeni bir meta sezgisel optimizasyon yöntemi olan Denge Optimizasyon (DO) algoritması ile aşı üretiminde gerçekleştirilen biyolojik fermantasyon işleminde kullanılan karıştırıcı modelleri için PID kontrolörler tasarlanmıştır. Öncelikle Denge Optimizasyon algoritması PID kontrolör parametrelerini optimize edebilecek kabiliyete ulaştırılmıştır. Daha sonra algoritmanın çalışma performansına etki eden parametrelerde çalışma esnasında deneysel olarak ayarlanmıştır. Özellikle literatürde daha önceden karıştırıcı modelleri için analitik yöntemlerle tasarlanmış olan PID kontrolör yerine performansı daha iyi olan Denge Optimizasyon algoritmasıyla tasarlanmış PID kontrolörlerin kapalı çevrim sonuçları karşılaştırmalı olarak sunulmuştur. Bu sayede yeni bir algoritma olan denge optimizasyon algoritmasının gerçek mühendislik problemlerinde de kullanılabileceği ve analitik yöntemlere karşın daha iyi kontrol performansına sahip PID kontrolörler türetilebileceği gösterilmektedir.

Anahtar Kelimeler: *Denge optimizasyon, Biyolojik fermentasyom, PID, Kontrolör*

Controller Design with Equilibrium Optimization Algorithm for an industrial biological fermentation process

ABSTRACT: In this study, PID controllers are designed for the mixer mathematical models for biological fermentation process in vaccine production with the Equilibrium Optimization (EO) algorithm, which is a new meta-heuristic optimization method. Firstly, Equilibrium Optimization algorithm has been provided with the ability to optimize PID controller parameters. Then, the parameters affecting the working performance of the algorithm were found experimentally during the study. In the literature, the closed loop results of PID controllers designed with the Equilibrium Optimization algorithm, which was previously designed with analytical methods for mixer system, have been presented comparatively. Thus, it is shown that the Equilibrium Optimization algorithm, which is a new algorithm, can also be used in real engineering problems and PID controllers with better control performance can be derived despite analytical methods.

Key Words: *Equilibrium optimizer, biological fermentation, PID, controller*

GİRİŞ (INTRODUCTION)

Günümüzde optimizasyon algoritmaları birçok mühendislik probleminde kullanılır. Bilindiği gibi bir problemin çözümünde kullanılan metotların yapısına problemin kendisi karar verir. Örneğin bir optimizasyon probleminde probleme ilişkin matematiksel modelin, kısıtların vs. tamamıyla bilinmesi durumunda analitik optimizasyon yöntemleri kullanılabilir. Analitik optimizasyon yöntemleri kendi arasında iki temel gruba ayrılabilir. Bunlar doğrusal programlama ve doğrusal olamayan

programlamadır. Problem çözümünde hangi analitik yöntemin kullanılacağına da problemin doğrusal veya doğrusal olmama durumu belirler (Bazaraa, Sherali, and Shetty 2005; Rabani 2007).

Analitik yöntemlerde karmaşık matematiksel yapılar kullanılır. Fakat bu yöntemlerin en önemli avantajı ise bulunan sonucun global veya lokal çözüm olmasının analizleri kesin olarak yapılabilir. Örneğin doğrusal olmayan bir optimizasyon probleminde problemin konveks veya konkavlığına göre Karush Khun Tucker ve Fritz John testleriyle bulunan sonuçların lokal veya global çözüm olma durumları kesin bir şekilde analiz edilebilir (Kuhn and Tucker 2014).

Fakat gerçek mühendislik problemlerinde ise problemin, matematiksel modelinin, kısıtlarının vs. tam anlamıyla bilinmesi son derece güçtür. Hatta model ve kısıtlar tamamıyla bilinmesi durumunda sistem, performansını etkileyen birçok bozucu ve dış etkene maruz kalabilir. Bu etkenlerde sistem modellerini değiştirir. Bundan dolayı mühendislik problemlerinin çözümünde genellikle nümerik optimizasyon yöntemleri kullanılır (Zhang and Ordóñez 2012). Nümerik optimizasyon yöntemleri deterministik ve stokastik yöntemler olarak iki grupta incelenebilir. Deterministik yöntemler parametre vektör uzaylarını deterministik matematiksel alt yapıya göre tarar. Stokastik yöntemler ise parametre vektör uzaylarını stokastik olarak tarar (Lin, Tsai, and Yu 2012). Özellikle mühendislik problemlerinde daha önce bahsedilen belirsizliklerinde dolayı stokastik optimizasyon algoritmaları kullanılır.

Literatürdeki stokastik optimizasyon algoritmaları doğadan esinlenen, insandan esinlenen ve fiziksel olaylardan esinlenen algoritmalar olarak sınıflandırılabilir. Doğadan esinlenen algoritmalar doğada yaşayan canlıların davranışlarında esinlenen algoritmalar olarak adlandırılabilir. Bu algoritmalara örnek karınca kolonisi (Dorigo and Socha 2007), yapay arı kolonisi (Karaboga and Akay 2009), guguk kuşu arama algoritması (Gandomi, Yang, and Alavi 2013), gri kurt optimizasyon algoritması (Mirjalili, Mirjalili, and Lewis 2014), salp sürüsü optimizasyon algoritması (Mirjalili et al. 2017) gösterilebilir. Fiziksel olaylardan esinlenen algoritmalara yer çekimi arama algoritması (Rashedi, Nezamabadi-pour, and Saryazdi 2009), yüklü sistem araması (Kaveh and Talatahari 2010) örnek olarak gösterilebilir. İnsandan esinlenen algoritmalara ise öğrenme ve öğretme tabanlı optimizasyon algoritması (Rao, Savsani, and Vakharia 2011), emperyalist yarışma algoritması (Atashpaz-Gargari and Lucas 2007) örnek olarak gösterilebilir.

Denge optimizasyon algoritması hem dinamik hem de denge durumlarını tahmin etmek için kullanılan hacim kütle denge kontrol modellerinden esinlenmektedir. Bu algoritma Faramarzi ve arkadaşları tarafından 2020 yılında önerilmiştir (Faramarzi et al. 2020). Algoritma süreçlerini stokastik olarak gerçekleştirir. Algoritmada her parçacık birer parametre vektör uzayını tarayan bir hareket olarak adlandırılır. DO algoritmasının temel makalesinde, literatürdeki PSO (Parçacık Sürü Optimizasyonu), GWO (Gri Kurt Optimizasyon), GA (Genetik Algoritma), GSA (Yerçekimi Arama Algoritması), SSA, CMA–ES (Kovaryans Matris Uyarlaması ile Evrim Stratejisi), SHADE (Başarı Geçmişine Dayalı Diferansiyel Evrime göre Parametre Uyarlama), CMA-ES in yarı parametre optimizasyonun liner popülasyonun büyüklüğünün azaltılmasıyla elde edilen LSHADE SPACMA gibi yeni ve uzun süredir literatürde kullanılan etkin optimizasyon algoritmalarıyla karşılaştırılmış ve üstünlükleri gösterilmiştir (Faramarzi et al. 2020). Özellikle algoritma performans analizlerinde çözümü son derece güç olan komposit karşılaştırma fonksiyonları üzerinden de test edilmiştir. Hatta algoritma temel mühendislik test problemleri üzerinde de test edilmiş ve performans analizleri detaylarıyla yapılmıştır. Bu çalışmada ise algoritmanın gerçek bir mühendislik problemi olan kontrolör tasarımı probleminde kullanılabileceği aşı üretiminde kullanılan karıştırıcı modelleri üzerinden gösterilmiştir.

Kontrolör tasarımı önemli mühendislik problemlerinde biridir. Çünkü sistemin referans girişi takip etmesi için kontrolör kullanılması gerekir. Fakat kontrolör tasarımı analitik olarak çözülmesi zor bir mühendislik problemidir. Özellikle sistem modelinin veya kıtsılarının bilinmemesi ve bozucunun tahmin edilemediği durumlarda zorluğun derecesi daha da artmaktadır. Bundan dolayı kontrolör tasarımında meta-sezgisel yöntemlerin kullanıldığı birçok çalışma bulunur. Örneğin (Ateş and Yeroglu 2016) çalışmasında modifiye edilmiş tabu arama algoritmasıyla PID ve kesir dereceli PID'yi sistem modelleri için tasarlanmıştır. Gerçek zamanlı uçuş kontrol sistemlerine kesir dereceli ve tamsayı dereceli kontrolörlerin tasarlandığı çalışma (Yeroglu ve Ateş 2014)'da bulunmaktadır. Modifiye edilmiş yapay

fizik optimizasyon algoritmasıyla sistemlere kontrolörlerin tasarlandığı çalışma (Ateş and Yeroğlu 2018)'de bulunmaktadır. Görüldüğü gibi meta sezgisel yöntemlerle kontrolör tasarımı literatür de geçerliliğini koruyan güncel bir çalışma alanıdır.

Fakat, kontrolör tasarımında önemli bir durum ise kontrolör tasarlanan modelin gerçek hayattan esinlenen bir model olması kullanılan algoritmanın problem çözümündeki etkinliğini de belirler. Özellikle biyolojik veya kimyasal reaksiyonların modellendiği çalışmalara etkin kontrolörler tasarlamak gerçek zamanlı üretim süreçlerine katkılar sunabilir. Bunda dolayı bu çalışmada aşı üretiminde gerekli olan fermentasyon sürecine ilişkin karıştırıcı ve çözünen oksijen miktarı arasındaki ilişkinin kurulduğu modeller kullanılmıştır (Khan et al. 2018).

(Khan et al. 2018) çalışmasında elde edilen veri setlerine göre oluşturulmuş modeller alınarak daha önce gerçek mühendislik probleminde kullanılmamış olan denge optimizasyon algoritmasıyla PID kontrolör tasarlanmıştır. Khan ve arkadaşları çalışmalarında önerdikleri model için doğrusal olmayan en küçük kareler yöntemiyle kontrolör tasarlamışlardır (Khan et al. 2018) . Fakat bilindiği gibi bu tarz analitik yöntemler sabit sistem şartlarında ve modelin tamamıyla bilindiği durumlarda etkin çalışmaktadır. Sistemsel değişikliklere karşın tepkileri yavaş veya yetersiz olabilir. Bunda dolayı bu çalışmada meta sezgisel bir yöntemle bu tarz bir sistem modeline kontrolör tasarımının yapılabileceğinin gösterilmesi bu alanda oluşturulan diğer modeller içinde meta sezgisel yöntemler ile kontrolör tasarlanabileceğini gösterebilir. Çalışmada dört adet sistem modeli kullanılmıştır. Öncelikle denge optimizasyon algoritması PID kontrolör parametrelerine optimize edebilecek hale getirilmiştir. Dört farklı fermentasyon işlemindeki karıştırıcı ve çözünen oksijen miktarı arasındaki ilişkiyi gösteren modeller için optimizasyon algoritması birbirinden bağımsız olarak çalıştırılmıştır. Her model için bulunan kontrolör parametreleri diğer modeller üzerinde de denemiş ve sonuçlar doğrusal olmayan en küçük kareler metodu sonuçlarıyla karşılaştırmalı olarak sunulmuştur. Çalışma sonucunda denge optimizasyon yönteminin mühendislik problemlerinde kullanılabileceği gösterilmiştir. Bunun yanı sıra gerçek bir mühendislik probleminden elde edilen ve aşı üretimindeki fermentasyon işlemi için karıştırıcı hızıyla çözünen oksijen miktarı arasındaki ilişkiye gösteren modeller için PID kontrolörler tasarlanmıştır. Bulunan kontrolörlerin mevcut durumdan daha iyi olduğu gözlemlenmiştir. Bu sayede bu tarz sistem değişikliği fazla olan özellikle bozucu oranı fazla olan sistemler için meta sezgisel yöntemlerle kontrolörler tasarlanabileceği gösterilmiştir.

DENGE OPTIMIZASYON ALGORİTMASI (EQULIBRIUM OPTIMIZATION ALGORITHM)

Algoritmanın Çıkış Terminolojisi (Inspiration)

Denge optimizasyon algoritmasının ilham kaynağı kontrol hacmi üzerindeki dinamik kütle dengesinin karşılaştırılmasıdır. Kütle denge denklemi, reaktif olmayan bir bileşenin konsantrasyonunu tanımlamak için kontrol hacmi içerisindeki çeşitli kaynakların ve sızıntı mekanizmasının bir fonksiyonu olarak kullanılır. Kütle denge denklemi kütlenin korunumu kuralını bir kontrol hacmi içerisinde sağlar. Aşağıda verilen birinci dereceden adi diferansiyel denklem genel kütle denge denklemini gösterir. Bu denkleme göre kütlenin zamana bağlı değişimi sisteme giren toplam kütleden sistemden çıkan toplam kütlenin farkının alınmasıyla elde edilir (Faramarzi et al. 2020). Zamana bağlı kütle değişim denklemi aşağıdaki gibidir:

$$V\frac{dC}{dt} = QC_{eq} - QC + G \tag{1}$$

Denklemde ki C kontrol hacmi içerisindeki konsantrasyonu (V), $V\frac{dC}{dt}$ ise kontrol hacmi içerisindeki kütlenin değişim oranını verir, Q volümetrik akış oranını, C_{eq} denge durumundaki konsantrasyonu, G ise kontrol hacmi içerisindeki kütle değişim oranını gösterir (Faramarzi et al. 2020). Denklemde ki $V\frac{dC}{dt}$ oranı sıfıra gittiği zaman sistem kalıcı denge durumuna ulaşır.

Q/V 'nin bir fonksiyonu olan dc/dt'nin çözülmesi için Denklem 1 yeniden düzenlenir. Denklemdeki Q/V yerleşme zamanının tersini veya devir hızını gösterir. Denklemde bu değer yerine λ yazılmıştır. Sonuç olarak Denklem 1 zamanın bir fonksiyonu olarak kontrol hacmi içerisindeki konsatrasyonun (yoğunluk) hesaplanması için aşağıdaki gibi yeniden oluşturulur (Faramarzi et al. 2020).

$$\frac{dC}{\lambda C_{eq} - \lambda C + \frac{G}{V}} = dt \tag{2}$$

Denklemde her iki tarafın integrali alınır:

$$\int_{C_0}^{C} \frac{dC}{\lambda C_{eq} - \lambda C + \frac{G}{V}} = \int_{t_0}^{t} dt \tag{3}$$

Sonuç olarak Denklem 3 aşağıdaki gibi düzenlenir:

$$C = C_{eq} + (C_0 - C_{eq})F + \frac{G}{\lambda V}(1 - F) \tag{4}$$

Denklem 4 deki F aşağıdaki gibi hesaplanır:

$$F = \exp[-\lambda(t - t_0)] \tag{5}$$

Denklemlerdeki t_0 ve C_0 sırası ile başlangıçtaki zaman ve konsantrasyon (yoğunluk) değerlerini gösterir. Denklem 4 ile bu dönüş hızına göre kontrol hacmi içerisindeki konsantrasyon hesaplanır. Veya Denklem 4 ile ortalama dönüş hızı doğrusal regresyon ile hesaplanır. Bu denklem Denge Optimizasyon algoritmasının temel denklemidir (Faramarzi et al. 2020). Denklem 4 Parçacık sürü optimizasyon algoritmasındaki gibi parçacıkların güncellenme kuralını gösterir. Denklem 4'de parçacıkların güncellenmesi için üç durum veya aşama bulunur. Her parçacık konsantrasyonun güncellenmesi için birbirinden bağımsız bu üç duruma göre çalışmaktadır.

Birinci aşama konsantrasyonun dengesidir. Bu aşama havuzda rastsal olarak seçilen en iyi çözümlerden birini ifade eder. Bu yapı denge havuzu olarak adlandırılır. Algoritmadaki havuz en iyi çözüm etrafında oluşturulan dört en iyi çözümü ve onların ortalamasından oluşur (Faramarzi et al. 2020). İkinci aşama mevcut parçacık ile denge durumu arasındaki konsantrasyonun (yoğunluk) farkını gösterir. Bu yapı doğrudan arama mekanizmasını etkiler.

Üçüncü aşama ise türetme oranı ile ilgilidir. Bu aşama aramanın iyileştirilmesinde rol oynar. Özellikle küçük adımlar veya oranlarla hareket edildiğinde arama sürecine önemli katkıları olur. Bahsedilen bu üç aşama arama performansını doğrudan etkiler. Algoritmanın arama sürecine etki eden yapıları aşağıda açıklanmaktadır (Faramarzi et al. 2020).

Başlangıç Değer ve Fonksiyonun Değerlendirilmesi (Initilialization and Function Evaluation)

Diğer meta-sezgizel optimizasyon algoritmalarında olduğu gibi DO (Denge Optimizasyon) algoritmasının optimizasyon sürecini başlatmak için başlangıç popülasyonu kullanılır. Başlangıç konsantrasyonu parçacık sayısına göre uniform rastsal dağılım kullanılarak oluşturulur. Başlangıç popülasyonu aşağıdaki gibidir:

$$C_i^{initial} = C_{min} + rand_i(C_{max} - C_{min}) \;\; i = 1, 2, 3, ...n \tag{6}$$

$C_i^{initial}$ her parçacık için başlangıç konsantrasyon vektörüdür. C_{max} ve C_{min} parçacıların maksimum ve minimum değerlerini gösterir. $rand_i$ [0 1] arasında türetilen rastsal vektördür. n ise popülasyondaki parçacıkların sayısını gösterir. Parçacıkların kendi amaç fonksiyon (fitness) değerine göre değerlendirilir ve daha sonra bu değer denge adaylarının belirlenmesinde kullanılır (Faramarzi et al. 2020).

Denge havuzu ve adaylar (Equilibrium Pool and Candidates (C_{eq}))

Denge durumu algoritmanın final durumu olarak gösterilir. Bu aşama algoritma içerisinde global optimal nokta olarak adlandırılır. Optimizasyon sürecinin başlangıcında denge durumu hakkında bilgi bulunmamaktadır. Sadece parçacıklar arama işlemini başlatmak için denge adaylarını kullanır. Farklı şartlar altında farklı tip problemlerin çözümünde elde edilen tecrübelere göre tüm optimizasyon sürecinde bu aday çözümler en iyi çözüme yakın olan dört farklı çözümdür. Yani en iyi çözümün etrafında dört adet aday çözüm belirlenir. Bu çözümlere ek olarak dört adet aday çözümün ortalaması da kullanılır. Bu dört aday DO algoritmasına arama kabiliyeti artırır. Optimizasyon problemine göre aday çözümlerin sayısı istenilen sayıda alınabilir.

Aslında en iyi çözüm etrafında belirli bir sayıda aday çözüm belirlemek literatürdeki başka çalışmalarda da mevcuttur (Mirjalili, Mirjalili, and Lewis 2014). Gri Kurt optimizasyon algoritmasında da üç aday çözüm kullanılır. Fakat dört aday çözümden fazla aday çözüm algoritma performansına olumsuz yönde etki ettiği literatürdeki birçok çalışmada açıklanmıştır. Algoritmada denge havuzu diye adlandırılan $C_{eq,pool}$ beş elemandan oluşur (Faramarzi et al. 2020).

$$\vec{C}_{eq,pool} = \left\{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\right\} \tag{7}$$

Her iterasyondaki her parçacığın konsantrasyonu aynı olasılıkla aday çözümler içersinde rastsal olarak seçilerek güncellenir. Örneğin birinci iterasyonda ilk parçacık tüm konsantrasyonun $\vec{C}_{eq(1)}$ göre güncellenir. Buna bağlı olarak ortalama konsantrasyonun $\vec{C}_{eq(ave)}$ değeri de güncellenir. Bundan dolayı optimizasyon süreci sonunda tüm parçacıklar tüm aday çözümlere göre aynı sayıdaki güncellenmeyle güncellenir.

Üssel Dönem (Exponential Term (F))

Ana konsantrasyon değerlerinin güncellendiği dönem üssel aşamasıdır. Denge optimizasyon algoritmasının algoritmik süreç içerisinde arama ve işletme (çalıştırma) durumları arasında bir dengeye sahiptir. Çünkü değişim hızı oranı (turnover rate) gerçek bir kontrol hacmi içerisinde zamana bağlı olarak değişebilir. λ [0,1] arasında rastsal bir vektörü gösterir. Bu vektör aşağıdaki gibidir (Faramarzi et al. 2020):

$$\vec{F} = e^{-\vec{\lambda}(t-t_0)} \tag{8}$$

Denklemdeki t fonksiyondaki iterasyon sayısını tanımlar. Bunda dolayı Denklem 9'da gösterildiği gibi iterasyon sayısına göre azalır ve t aşağıdaki gibi hesaplanır.

$$t = \left(1 - \frac{Iter}{Max\_Iter}\right)^{\left(a_2 \frac{Iter}{Max\_Iter}\right)} \tag{9}$$

Denklemdeki Iter ve Max_Iter mevcut ve maksimum iterasyon sayılarını gösterir. a_2 sabit bir değerdir ve kullanım kabiliyetini gösterir. Yapılan işlemler sonucunda arama ve çalıştırma (kullanma) kabiliyeti artırılarak algoritmanın optimal noktaya ulaşmasının sağlamak için Denklem 10 göz önünde bulundurulur (Faramarzi et al. 2020).

$$\vec{t}_0 = \frac{1}{\vec{\lambda}} \ln(-a_1 sign(\vec{r} - 0.5)[1 - e^{-\vec{\lambda}t}]) + t \tag{10}$$

Denklemde ki a_1 arama kabiliyetini kontrol eden sabit bir sayıdır. a_1 değerinin büyük olması arama kabiliyetini artırırken kullanma ve çalıştırma kabiliyetini de azaltır. Benzer şekilde a_2'nin yüksek olması çalıştırma veya kullanma kabiliyetini artırırken arama kabiliyetini azaltır. Denklem 10 'daki üçüncü bileşen olan (r-0.5) arama ve kullanma durumlarına doğrudan etki eder. Çalışmada r, 0 ile 1 arasında

rastsal bir vektördür. Algoritmada karşılaştırma fonksiyonu testleri için a_1=2 ve a_2=1 değerleri kullanılmıştır. Bu değerler ampirik testler neticesinde bulunmuştur (Faramarzi et al. 2020). Fakat başka problemlerde bu değerlerin değiştirilmesi gerekebilir (Faramarzi et al. 2020). Denklem 10 Denklem 8'de yerine konulmasıyla Denklem'11 elde edilir.

$$\vec{F} = a_1 \text{sign}(\vec{r} - 0.5)[e^{-\vec{\lambda}t} - 1] \tag{11}$$

Türetme Oranı (Generation Rate (G))

Türetme oranı DO algoritmasının en önemli yapılarında birdir. Bu oran algoritmanın çalışma veya kullanma yeteneğini geliştirerek tam çözümün bulunmasını sağlar. Birçok mühendislik uygulamalarında türetme oranı için birçok model kullanılır (Guo 2002).Örneğin bunlardan biri olan birinci dereceden üstsel azalma modeli aşağıdaki gibi tanımlanır (Faramarzi et al. 2020).

$$\vec{G} = \vec{G}_0 e^{-\vec{k}(t-t_0)} \tag{12}$$

Denklemdeki G_0 başlangıç değerini gösterir. k azalma katsayısıdır. Algoritmanın arama yapma sürecinin kontrol edilmesi için bu çalışmada $k = \lambda$ olarak alınmıştır. Türetme oranının son denklemi ise aşağıdaki gibidir.

$$\vec{G} = \vec{G}_0 e^{-\vec{k}(t-t_0)} = \vec{G}_o \vec{F} \tag{13}$$

Denklemdeki G_0 aşağıdaki gibi hesaplanır:

$$\vec{G}_0 = \vec{GCP}(\vec{C}_{eq} - \vec{\lambda}\vec{C}) \tag{14}$$

$$\vec{GCP} = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \tag{15}$$

Denklemdeki r_1 ve r_2 değerleri [0,1] aralığındaki rastsal değerlerdir. GCP Türetme oranı kontrol parametresidir. Bu parametre güncellenme süreçlerine katkı sağlamak için türetme olasılıklarını içerir. Bu olasılık bilgisi parçacıkların kendi durumlarını güncellerken kaç adet türetme durumu ve aşamasının kullanıldığını belirler. Bu yapı türetme olasılığı (GP- Genaration Probability) olarak adlandırılır. Bu mekanizma Denklem 14 ve Denklem 15'e göre türetilir. Örneğin GCP'nin sıfıra eşit olması G'nin sıfıra eşit olması anlamına gelir. Bu da tüm yönlerdeki parçacıkların türetme oranı kullanmadan güncellendiği anlamına gelir. Arama ve kullanma veya çalıştırma arasında denge olması için GP=0.5 olarak alınmıştır. Sonuç olarak Denge optimizasyon algoritmasının güncellenme kuralları aşağıdaki gibidir (Faramarzi et al. 2020):

$$\vec{C} = \vec{C}_{eq} + (\vec{C} - \vec{C}_{eq}).\vec{F} + \frac{\vec{G}}{\vec{\lambda}\vec{V}}(1 - \vec{F}) \tag{16}$$

Denklem 16'daki birinci aşama konsantrasyonun dengesidir. İkinci ve üçüncü aşamalar ise konsantrasyonun değişimidir. İkinci aşama algoritmada global optimum noktanın bulunmasından sorumludur. Bu aşama daha çok arama fazına etki eden bir aşamadır. Bu aşamada, doğrudan denge konsantrasyon değerinden mevcut konsantrasyon yani parçacıktaki değerin farkı alınır. İkinci aşamada global nokta bulunduktan sonra üçüncü aşamada ise bulunan sonuçların doğruluğu denetlenir. Algoritma temelde şöyle çalışmaktadır. Başlangıç popülasyonu oluşturulduğunda algoritma büyük rastsal değerlere göre hareket eder. Bu adım güncellenme prosedürü ile parçacıkların optimal noktaya yerleşmesini sağlar. Bu durumun ters senaryosu da geçerlidir. Eğer değerler yani başlangıç popülasyonu optimal noktalara yakın ise rastsal değerler türetilirken adımlar küçültülmelidir. Bu yapı algoritmaya n boyutlu ve yönlü aday popülasyonlarla çalışma fırsatı verir (Faramarzi et al. 2020).

Parçacık Hafızasının Kaydedilmesi (Particle Memory Saving)

Bu bölümde parçacıkların parametre vektör uzayındaki yerlerinin ve amaç fonksiyon değerleri kaydedilir. Bu yapı PSO algoritmasındaki en iyi değer ile benzerlik taşır. Mevcut durumdaki ve bir önceki durumdaki amaç fonksiyon değerleri karşılaştırılır. Eğer amaç fonksiyonu maksimize ve minimize

edilecek eğilimde ise mevcut değer en iyi değerin üzerine yazılarak güncellenir. Şekil 3 de bu yapı gösterilmektedir (Faramarzi et al. 2020).

Denge optimizasyon algoritmasının arama kabiliyeti (Exploration ability of EO)

Çalışmanın bu bölümde Denge Optimizasyon Algoritmasının çalışma mekanizmasına etki eden parametreler detaylı olarak açıklanmaktadır:

a_1: Tarama yapısını kontrol eden bir katsayıdır. Parçacığın yeni pozisyonunun denge aday çözümünden ne kadar uzakta olduğunu belirler. Önceki bölümlerde bahsedildiği gibi a_1 değerinin büyük olması yüksek arama kabiliyeti sağlar. Fakat değerin üçten büyük olması arama performansını azaltır. Çünkü a_1 konsantrasyon değişimini büyütebilir. Bunda dolayı bu değer arama kabiliyetini artırmak için olması gerektiği kadar büyük olmalıdır. Bu değerin üç ile sınırlandırılması aslında ampirik olarak elde edilmiştir. Bu tarz diğer meta-sezgisel optimizasyon algoritmalarda da sınırlandırma mevcuttur. Örneğin PSO'da sosyal ve kavramsal parametrelerin dört'e eşit veya küçük olması şartı bulunur (Faramarzi et al. 2020).

Sing(r-0.5)=arama yönünü kontrol eder. r 0 ile 1 arasında rastsal bir değerdir.

Türetme Olasılığı (Generation Probability-(GP))

GP=1 olması durumunda optimizasyon sürecinde değişim olmadığını bu da yüksek oranda tarama kapasitesini gösterir. GP=0 olması durumunda ise değişim oranının optimizasyon sürecindeki aday çözümlere etki ettiğini gösterir. Ampirik testler sonucunda arama ve kullanma süreçlerinin dengeli olması için GP=0.5 şeklinde alınmıştır (Faramarzi et al. 2020).

Denge Havuzu (Equlibrium Pool)

Bu vektör beş elemanda oluşur veya beş parçacığa sahiptir. Bu beş parametrenin seçimi bazen keyfi bazen de ampirik testlere göre yapılır. Başlangıç iterasyonun da tüm aday çözümler belirli bir mesafeyle birbirinde uzaktır. Bu aday çözümler göre konsantrasyon yani amaç fonksiyon değerinin güncellenmesiyle parametre vektör uzayında algoritmanın arama yapması daha doğrusu global noktanın bulma prosedürünü artırır.

Denge optimizasyon algoritmasının kullanma ve çalıştırma kabiliyeti (Exploitation ability of EO)

Bu aşamada algoritmanın kullanma ve lokal arama aşamaları gerçekleştirilir. Bu aşamaya etki eden parametreler aşağıdaki gibidir (Faramarzi et al. 2020).

a_2: Bu parametre a_1 parametresi gibidir. Fakat a_2 parametresi kullanma veya deneme fazını kontrol eder. Kullanma durumunun genliğini belirler bu sayede en iyi çözüm etrafında denemeler yapılmasını sağlar.

Hafızanın Kaydedilmesi (Memory Saving)

İlgili popülasyonda bulunan parçacıkların en iyi ve en kötü değerlerini kaydeder. Bu yapı optimizasyon algoritmasında kullanılan kullanma sürecini doğrudan etki eder.

Denge Havuzu (Equliibrium Pool)

Parametrelerin yinelenmesiyle birlikte arama fazı kaybolur deneme fazı etkin hale gelir. Bundan dolayı her iterasyonda denge adaylarının birbirine yakın olması durumunda konsantrasyon güncelleme prosedürü aday çözümler etrafında lokal aramalar yapmaya yardım eder. DO algoritmasının sözde kodu aşağıda bulunmaktadır.

Başlangıç popülasyon parçacıklarının belirlenmesi

Denge aday çözümlerin maliyet fonksiyonlarının büyük bir sayı ile atanması
Algoritmanın parametrelerin atanması a_1=2, a_2=1; GP=0.5.
While Iter<Max_Iter
for i=1:Parçacık Sayısı(n)
Her parçacığın maliyet fonksiyonun hesaplanması
if $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(1)})$
$\vec{C}_{eq1}$ 'i ile $\vec{C}_i$ ve $\text{fit}(\vec{C}_{eq(1)})$ 'i $\text{fit}(\vec{C}_i)$ ile değiştir
Elseif $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(1)})$ and $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(2)})$
$\vec{C}_{eq2}$ 'i ile $\vec{C}_i$ ve $\text{fit}(\vec{C}_{eq(2)})$ 'i $\text{fit}(\vec{C}_i)$ ile değiştir
Elseif $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(1)})$ and $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(2)})$ and $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(3)})$
$\vec{C}_{eq3}$ 'i ile $\vec{C}_i$ ve $\text{fit}(\vec{C}_{eq(3)})$ 'i $\text{fit}(\vec{C}_i)$ ile değiştir
Elseif $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(1)})$ and $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(2)})$ and $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(3)})$ and $\text{fit}(\vec{C}_i) < \text{fit}(\vec{C}_{eq(4)})$
$\vec{C}_{eq4}$ 'ü ile $\vec{C}_i$ ve $\text{fit}(\vec{C}_{eq(4)})$ 'i $\text{fit}(\vec{C}_i)$ ile değiştir
End
End

$$\vec{C}_{ave} = \left(\vec{C}_{eq(1)} + \vec{C}_{eq(2)} + \vec{C}_{eq(3)} + \vec{C}_{eq(4)}\right)/4$$

Denge havuzunun oluşturulması $\vec{C}_{eq,pool} = \left\{\vec{C}_{eq(1)}, \vec{C}_{eq(2)}, \vec{C}_{eq(3)}, \vec{C}_{eq(4)}, \vec{C}_{eq(ave)}\right\}$
Hafıza kaydının tamamlanması (if Iter<1)
t nin tanımlanması $t = \left(1 - \frac{\text{Iter}}{\text{Max\_Iter}}\right)^{\left(a_2 \frac{\text{Iter}}{\text{Max\_Iter}}\right)}$
for i=1:parçacık sayısı (n)
Denge havuz vektöründe bir aday çözümün rastsal seçilmesi
Random vektörlerin Denklem 11 e göre türetilmesi $\vec{\lambda}, \vec{r}$ $\vec{F} = a_1 \text{sign}(\vec{r} - 0.5)[e^{-\vec{\lambda}t} - 1]$

$\vec{F} = a_1 \text{sign}(\vec{r} - 0.5)[e^{-\vec{\lambda}t} - 1]$ düzenlenmesi

$\vec{GCP} = \begin{Bmatrix} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{Bmatrix}$ düzenlenmesi

$\vec{G}_0 = \vec{GCP}(\vec{C}_{eq} - \vec{\lambda}\vec{C})$ düzenlenmesi

$\vec{G} = \vec{G}_0 e^{-\vec{k}(t-t_0)} = \vec{G}_o \vec{F}$ düzenlenmesi

Konsantrasyonun Güncellenmesi $\vec{C} = \vec{C}_{eq} + (\vec{C} - \vec{C}_{eq}).\vec{F} + \frac{\vec{G}}{\vec{\lambda}V}(1 - \vec{F})$

End
Iter=Iter+1
End

ENDÜSTRİYEL BİOLOJİK FERMANTASYON İŞLEMİ (INDUSTRIAL BIOLOGICAL FERMENTATION PROCESS)

Aşı üretimi birçok aşamadan oluşur. Bu aşamalardan ilki mikrobiyal fermentör içerisinde organizmanın yetiştirilmesidir. Optimal aşı sentezinin gerçekleşmesi için temel etken yeterli miktarda oksijen çözümünün kontrol edilmesidir. Çalışmanın bu bölümünde aşı üretimi için karıştırıcının hızının değişimiyle çözünen oksijen miktarının değişimi için oluşturulan veri setleri giriş ve çıkış verisi olarak kullanıldığı modeller kullanılmıştır (Khan et al. 2018). Khan ve arkadaşları dinamik bir fermenter sistemi oluşturmuşlardır. Bu çalışmada bu sisteme uygulanan giriş verilerine karşın çıkış verileri açık çevrim olarak elde edilmiştir. Elde edilen veri setini modellemek için birçok farklı model yapısı kullanılmıştır.

Bunlar içerisinde en iyi cevabı veren dört model seçilmiştir. Bu çalışmada da Khan ve arkadaşları tarafında elde edilmiş modeller için PID kontrolörler DO algoritmasıyla tasarlanmıştır.

G_P transfer fonksiyonu için karıştırıcının hızı U(s), çözünen oksijen miktarı ise Y(s) şeklindedir. Çalışmada modeller Matlab "tfest" modelleme araç kutusuna göre gerçekleştirilmiştir (Khan et al. 2018). Çalışmada önerilen modeller aşağıdaki tabloda verilmektedir.

Çizelge 1. Karıştırıcı hızına göre oksijen çözünme modelleri

Tablo 1. *Dissolved oxygen model according to stirrer rate*

Model	Modelin Türü	Kat sayılar	Modelin Transfer fonksiyonu
Model 1 G_{p1}	$\frac{Ke^{-\theta_d s}s}{(\tau_1 s+1)(\tau_2 s+1)}$	$K=2033;\tau_1=86;\tau_2=5656;\theta=24$	$\frac{2033s}{486416s^2+5742s+1}e^{-24s}$
Model 2 G_{p2}	$\frac{Ke^{-\theta_d s}(\tau_3 s+1)}{(\tau_1 s+1)(\tau_2 s+1)}$	$K=0.326;\tau_1=89;\tau_2=564.7;\tau_3=719.1;\theta=18$	$\frac{234.4266s+0.326}{50263s^2+653.7s+1}e^{-18s}$
Model 3 G_{p3}	$\frac{Ke^{-\theta_d s}(\tau_3 s+1)}{(\tau_1 s+1)(\tau_2 s+1)}$	$K=-0.829;\tau_1=80.7;\tau_2=11960;\tau_3=-5721;\theta=24$	$\frac{4742.7s-0.829}{965172s^2+12401s+1}e^{-24s}$
Model 4 G_{p4}	$\frac{Ke^{-\theta_d s}s}{(\tau_1 s+1)(\tau_2 s+1)}$	$K=2130;\tau_1=97.6;\tau_2=5461;\theta=7$	$\frac{2130s}{532990s^2+5558.6s+1}e^{-7s}$

DENGE OPTIMIZASYON ALGORTIMASIYLA KONTROLÖR TASARIMI VE SIMULASYON SONUÇLARI (CONTROLLER DESİGN WITH EQULIIBRIUM OPTIMIZATION METHOD AND SIMULATION RESULTS)

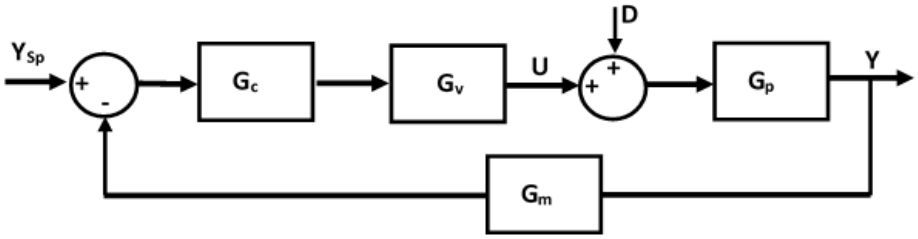
Aşı üretiminde karıştırıcının hızıyla çözünen oksijen miktarı arasında kolerasyonun olduğu bilinir. Çözünen oksijenin kontrolü için karıştırıcının hızının kontrol edilmesi gerekir. (Khan et al. 2018) çalışmasında dört farklı örnek için karıştırıcı hızıyla çözünen oksijen miktarı arasında ilişkinin olduğu model belirlenmiştir. Daha sonra bu modeller için doğrusal olmayan en küçük kareler optimizasyon metoduyla kontrolörler IAE hata yapısına göre tasarlanmıştır. Aslında dört farklı örnek için elde edilen modeller Çizelge 1'de verilmektedir. Bu modeller analitik olarak denetçi tasarlamak güçtür ve tasarlanan denetçileri kontrol performansları son derece iyidir. (Khan et al. 2018) çalışmasında karıştırıcı hızına karşın oksijen çözünüm sistem modelleri için doğrusal olmayan en küçük kareler optimizasyon algoritmasıyla tasarlanan PID denetçi parametreleri Çizelge 2'de verilmektedir. Bu değerler Çizelge 1'de verilen model 1 için IAE hata fonksiyonuna göre bulunmuş ve diğer model 2 model 3 ve model 4 için ise bu kontrolör doğrudan denenmiş ve sonuçları sunulmuştur.

Fakat bu çalışmada ise 2020 yılında önerilmiş olan ve daha önce her hangi gerçek bir mühendislik probleminde kullanılmamış olan denge optimizasyon algoritması kullanılmıştır (Faramarzi et al. 2020). Öncelikle algoritma PID kontrolör parametrelerini optimize edecek kabiliyete getirilmiş ve MATLAB ortamında kodlanmıştır. Çalışma esnasında Denge optimizasyon algoritmasının performansına etki edebilecek yapılarında analizi yapılmış ve bu problem için yeniden belirlenmiştir. Örneğin (Faramarzi et al. 2020) çalışmasında denge optimizasyon algoritması benchmark fonksiyonları üzerinde çalıştırılırken a_1=2 a_2=1 alınmasına karşın önceki bölümde detayları verilmiş olan bu parametreler bu çalışmada a_1=0.7, a_2=0.2 değerleri alınmıştır. Bu değerler ampirik olarak çoklu testler yapılara tespit edilmiştir. Optimizasyon esnasında parametreler tamamıyla serbest bırakılmış yani algoritma sınırlı bir alanda arama uzaylarını oluşturmamıştır. Bilindiği gibi PID kontrolör yapısında üç parametre bulunmaktadır.

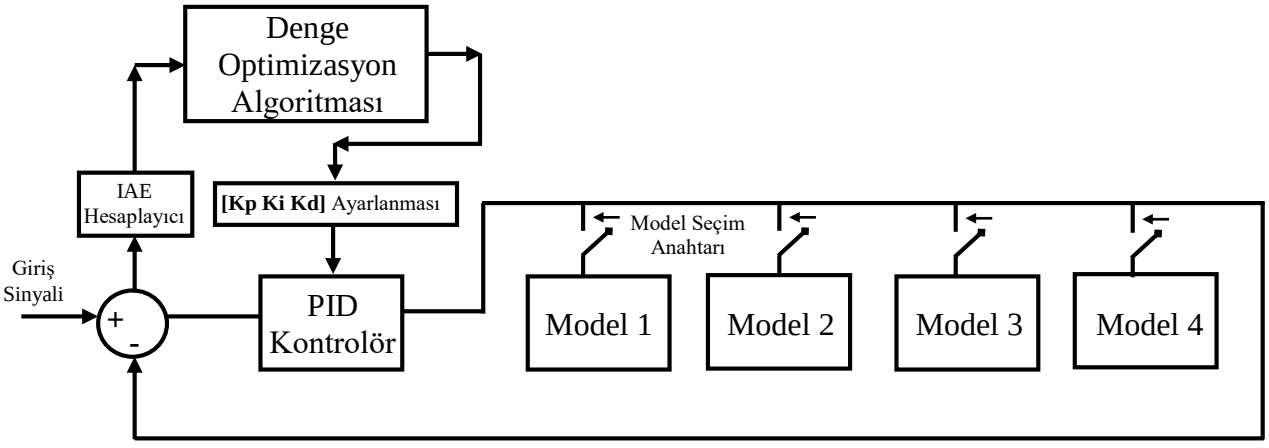
Bundan dolayı aslında arama uzayımız üç yönlü bir arama uzayıdır. Yani her parametre için denge optimizasyon algoritmasının felsefesinden dolayı dörtlü aday çözümler önerilmiştir. Bu aday çözümlere göre de aday havuzu oluşturulmuştur. Daha sonra popülasyondaki her konfigürasyonun amaç fonksiyon değerine bakılarak algoritma önceki bölümlerde verilen optimizasyon algoritmasının terminolojisini takip ederek çalıştırılmıştır.

Çalışma esnasında amaç fonksiyonu olarak (Khan et al. 2018) çalışmasında olduğu gibi IAE hata fonksiyon tanımı kullanılmıştır. Optimizasyon süresince kullanılan sistem model terminolojisi Şekil 1'de sunulmaktadır. Şekil 1'de optimizasyonda kullanılan geri beslemeli kontrol sistemi görülmektedir. Bu şekilde G_c PID kontrolörü gösterir. G_v final kontrol elemanı olarak adlandırılır ve 20 olarak alınmıştır. U karıştırıcı oranı olarak adlandırılır. G_p ise sistemin modelidir. Bu çalışmada dört örnek veri seti için dört farklı model Şekil 1'de gösterilen kapalı çevrim kontrol sisteminde G_p bloğu yerinde kullanılmıştır. D is bozucu girişidir. Optimizasyon uygulamasında bu değer sıfır alınmıştır. G_m ise sensör veya verici olarak adlandırılmakta bu çalışmada G_m ise 1 alınmıştır (Faramarzi et al. 2020).

Optimizasyon süresince öncelikle G_p yerine Model 1 yazılmış ve 10 iterasyon 15 parçacık sayısıyla IAE amaç fonksiyonu göre çalıştırılmış ve PID kontrol parametreleri türetilmiştir. Daha sonra aynı şartla altında sadece G_p değiştirilerek yerine model 2 yapısı konulmuştur ve aynı şekilde PID parametreleri tespit edilmiştir. Bu uygulama benzer şekilde model 3 ve model 4 içinde tekrarlanmıştır.



Şekil 1: Kontrol Yapısı (Khan et al. 2018)
Figure 1. *Control Structure*



Şekil 2: Optimizasyon Yapısı
Figure 2. *Hierarchical Optimization Structure*

Şekil 2'de çalışmada kullanılan algoritmik yapı sunulmuştur. Algoritmada optimizasyon süresince model seçim anahtarına göre model seçilmekte ve kontrolör parametreleri optimize edilmektedir. Örneğin birinci anahtar kapatıldığında model 1 seçilmekte model 1 için denge optimizasyon algoritması PID parametrelerini optimize etmek için belirlenen iterasyonlar sonucunda çalışmakta ve Çizelge 2'deki Model 1 için bulunan kontrol parametrelerini türetmektedir. Bu kontrolör parametreleri Model 1, Model

2, Model 3 ve Model 4 için denenerek ilgili modeller için doğrusal olmayan optimizasyon algoritması sonuçlarıyla karşılaştırılmıştır. Sonuçlar Şekil 3'de gösterilmektedir. Ayrıca her model için optimizasyon süresince hata fonksiyonun değişim grafiker gösterilmektedir. Örneğin Şekil 3'de Model 1 için optimizasyon süresince hata fonksiyonun azalan bir eğilimde olması optimizasyon algoritmasının performansının iyi olduğunu gösterir.

Bunların yanı sıra sonuçlardan da görüldüğü gibi optimizasyon algoritması analitik yolla tasarlanmış kontrolörden bile daha iyi kontrolörler tasarlayabilmektedir. Bu da optimizasyon algoritmasının diğer bir üstünlüğü olarak gösterilebilir. Yukarıda bahsedilen yapı diğer üç model içinde tekrarlanmış ve sonuçlar Şekil4, Şekil 5 ve Şekil 6'da sırasıyla sunulmaktadır.

Şekil 2'de verilen anahtar 1 kapatılarak algoritma Model 1 için çalıştırılmıştır. Çalışmada toplam popülasyon sayısı 10, parçacık sayısı 15 olarak belirlenmiştir. Çalışmada a_1=0.7 ve a_2=0.2'dir. Çizelge 1'de görüldüğü gibi Model 1 birinci dereceden zaman gecikmeli ve pay kısmında türev olan bir modeldir. Kontrolör tasarımında referans çalışmadaki olduğu gibi IAE hata değeri amaç fonksiyonu olarak kullanılmıştır (Khan et al. 2018). Tasarlanmış olan kontrolör parametreleri Çizelge 2'de sunulmuştur. Şekil 3 de Model 1 için tasarlanan kontrolörün Model 2 Model 3 ve Model 4'e göre giriş sinyaline karşın cevapları bulunmaktadır. Şekillerden de görüldüğü gibi DO algoritması rastsal noktalarda ve çok geniş bir parametre aralığında başlamasına rağmen doğrusal olmayan en küçük kareler optimizasyon yöntemiyle ile elde edilen kontrolör cevaplarına karşın daha iyi sonuçlar üretebildiği gözlemlenmiştir. Özellikle Şekil 1a, Şekil 1b ve şekil1c'de mevcut kontrolör cevaplarına karşın yerleşme zamanında iyileşmeler mevcut iken Şekil 1d ise hem yerleşme hem de yükselme zamanlarına karşı iyileşmenin olduğu tespit edilmiştir. Bunların yanı sıra şekil 1e'de ise Model 1 için yapılan optimizasyon süresince hatanın değişimi gösterilmektedir. Hata fonksiyonu minimize edilmesi gereken bir amaç fonksiyonudur. Bundan dolayı optimizasyon süresince azalması beklenir. Şekil 1 e 'de ise iterasyonlar süresince hata fonksiyonun sürekli azaldığı görülür. Bu da DO algoritmasının mühendislik problemlerinde özellikle kontrolör tasarımında etkin bir yöntem olduğunu gösterir.

Model 2 için optimizasyon algoritması model 1'de olduğu gibi aynı şartla altında çalıştırılmıştır. Şekil 4 de ise model 2 için tasarlanan kontrolörün diğer modeller için de cevapları karşılaştırmalı olarak sunulmuştur. Model 2 için elde edilen kontrolör parametreleri Çizelge 2'de bulunmaktadır. Özellikle Şekil 4b ve 4d daha iyi yerleşme zamanı ve yükselme zamanı cevapları üretmiştir. Görüldüğü gibi optimizasyon algoritması referans aldığı model için Model 1 de olduğu gibi Model 2 içinde mevcut durumdan daha iyi kontrolör parametreleri türetmiştir. Model 2 için optimizasyonla elde edilen kontrolörün sistem cevabı Şekil 4b'de bulunur. Diğer sonuçlar ise Model 2 için elde edilmiş kontrolörün diğer modellere olan cevaplarıdır. Bazı şekiller de sonuçların mevcut sistemden bir miktar kötü çıkması optimizasyon algoritmasının ilgili model için çalıştıramadığından dolayıdır. Hatta denge optimizasyon algoritmasıyla bulunan kontrolörler ilgili model için optimize edilmemiş sistem modellerine göre daha başarılı kontrol cevapları ürettiği şekillerde gözlemlenmiştir.

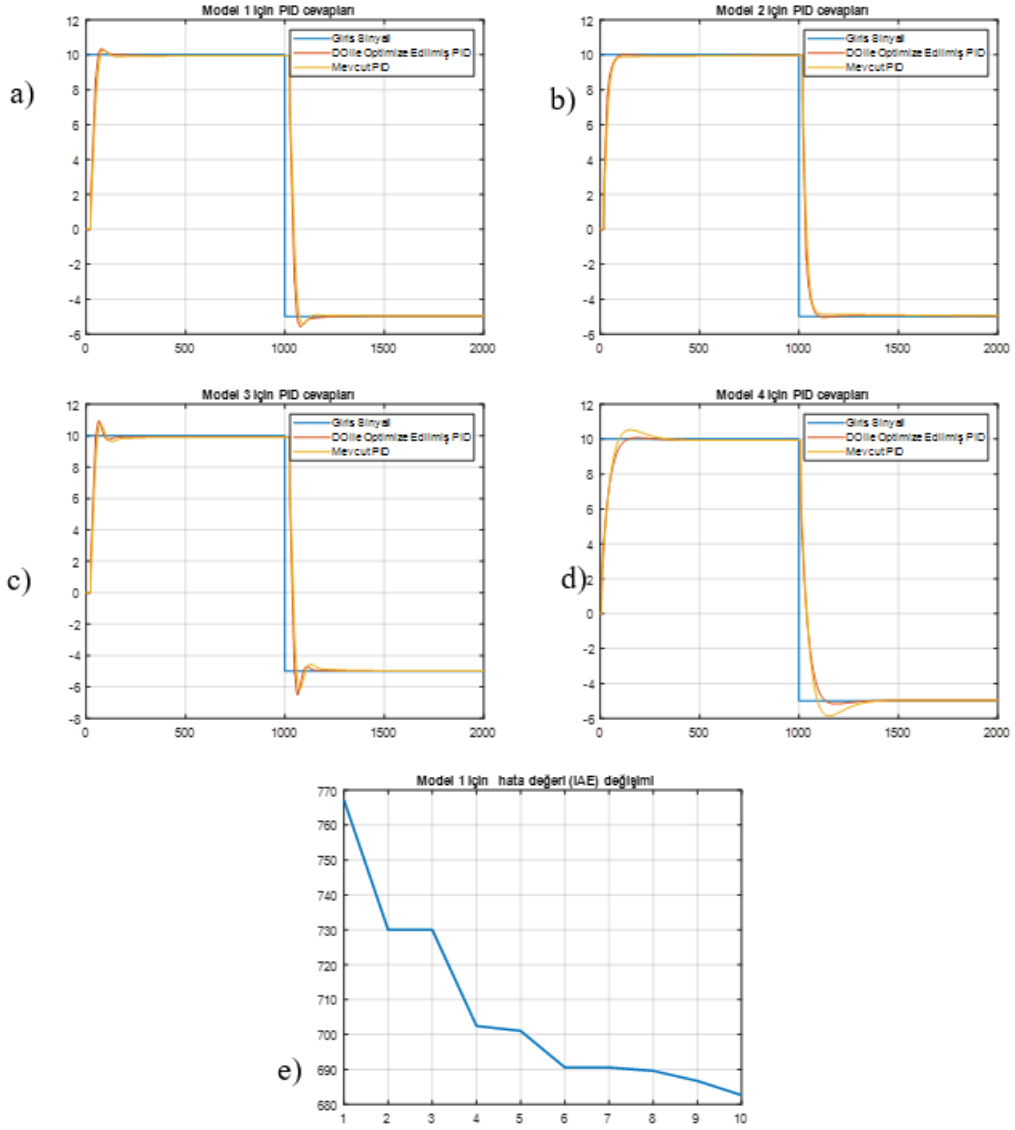
Model 3 ve Model 4 için aynı şartlar altında optimizasyon algoritmaları çalıştırılmış ve elde edilen kontrolör parametrelerine göre sistem cevapları Şekil 5 ve Şekil 6 da sunulmuştur. Şekillerden de görüldüğü gibi denge optimizasyon algoritması her durumda mevcut durumdan daha iyi kontrol cevapları üretmiştir. Özellikle yükselme ve yerleşme zamanlarındaki iyileşme açıkça görülmektedir. Bunların yanı sıra optimizasyon algoritmalarının performans göstergesi olan hatanın azalması da tüm modeller için gözlemlenmiştir. Şekil 5e ve şekil 6e de her iki model içinde hataların düşen eğilimde olduğu denge optimizasyon algoritmasın kontrolör tasarımı problemlerinde çalıştığının göstergesidir.

Elde edilen sonuçlara göre yani her model için optimizasyon algoritması bağımsız olarak çalıştırılmış ve her model için bulunan kontrolörler diğer modellerde de denenerek en uygun PID kontrolörün bulunması amaçlandığında dolayı tüm sonuçla karşılaştırmalı olarak sunulmuştur. Sistemlerin cevapları incelendiğinde Model 1 için elde edilen kontrolörlerin diğer sitemlerde daha iyi cevaplar verdiği görülmektedir.

Çizelge 2. DO optimizasyon algoritmasıyla elde edilen kontrolörler parametreleri

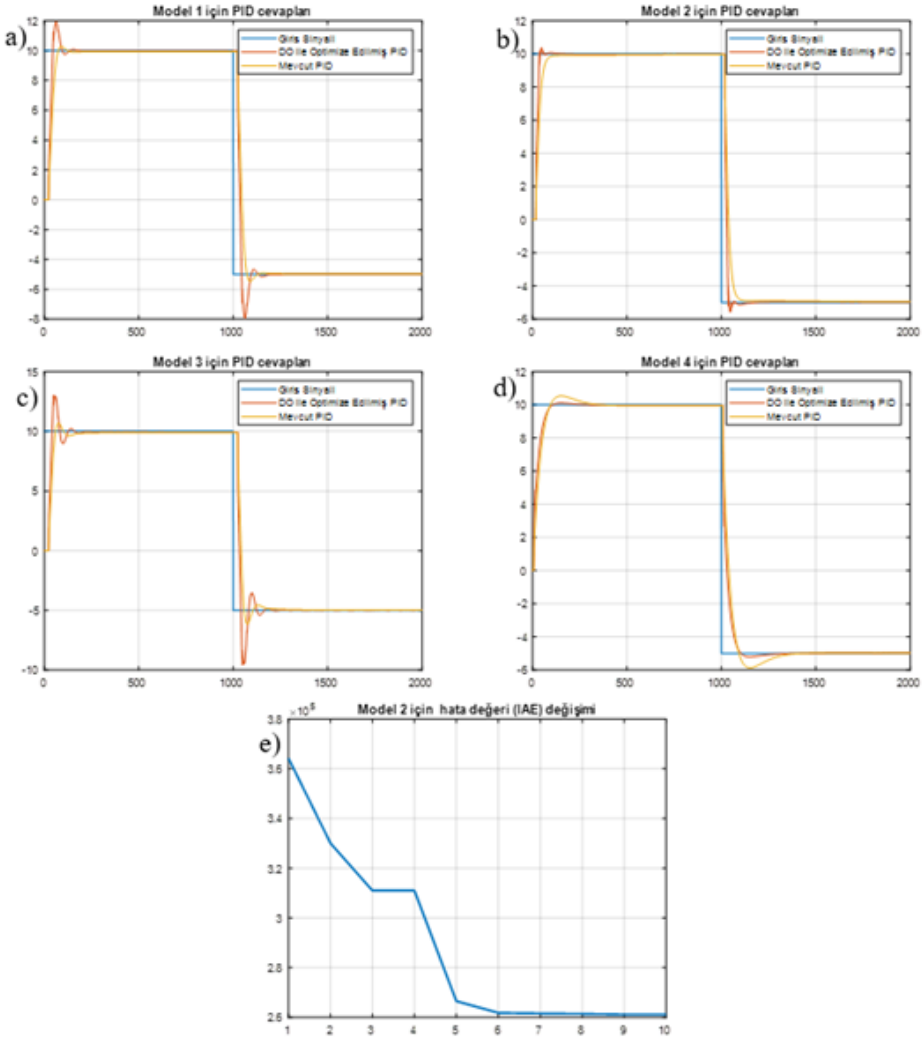
Tablo 2. *Obtained controller parameter with EO Algorithm*

Model	K_p	K_i	K_d
Model 1: $G_{p1} = \frac{2033s}{486416s^2+5742s+1}e^{-24s}$	0.3622867195	0.004124040288	2.324804293
Model 2: $G_{p2} = \frac{234.4266s+0.326}{50263s^2+653.7s+1}e^{-18s}$	0.4684828186	0.005401002796	3
Model 3: $G_{p3} = \frac{4742.7s-0.829}{965172s^2+12401s+1}e^{-24s}$	0.3394137491	0.004374472308	3
Model 4: $G_{p4} = \frac{2130s}{532990s^2+5558.6s+1}e^{-7s}$	0.4	0.004	2.077914127
Referans PID Kontrolör Katsayıları (Doğrusal Olamayan En Küçük Kareler Yöntemi)	0.3226	0.0035	1.3872



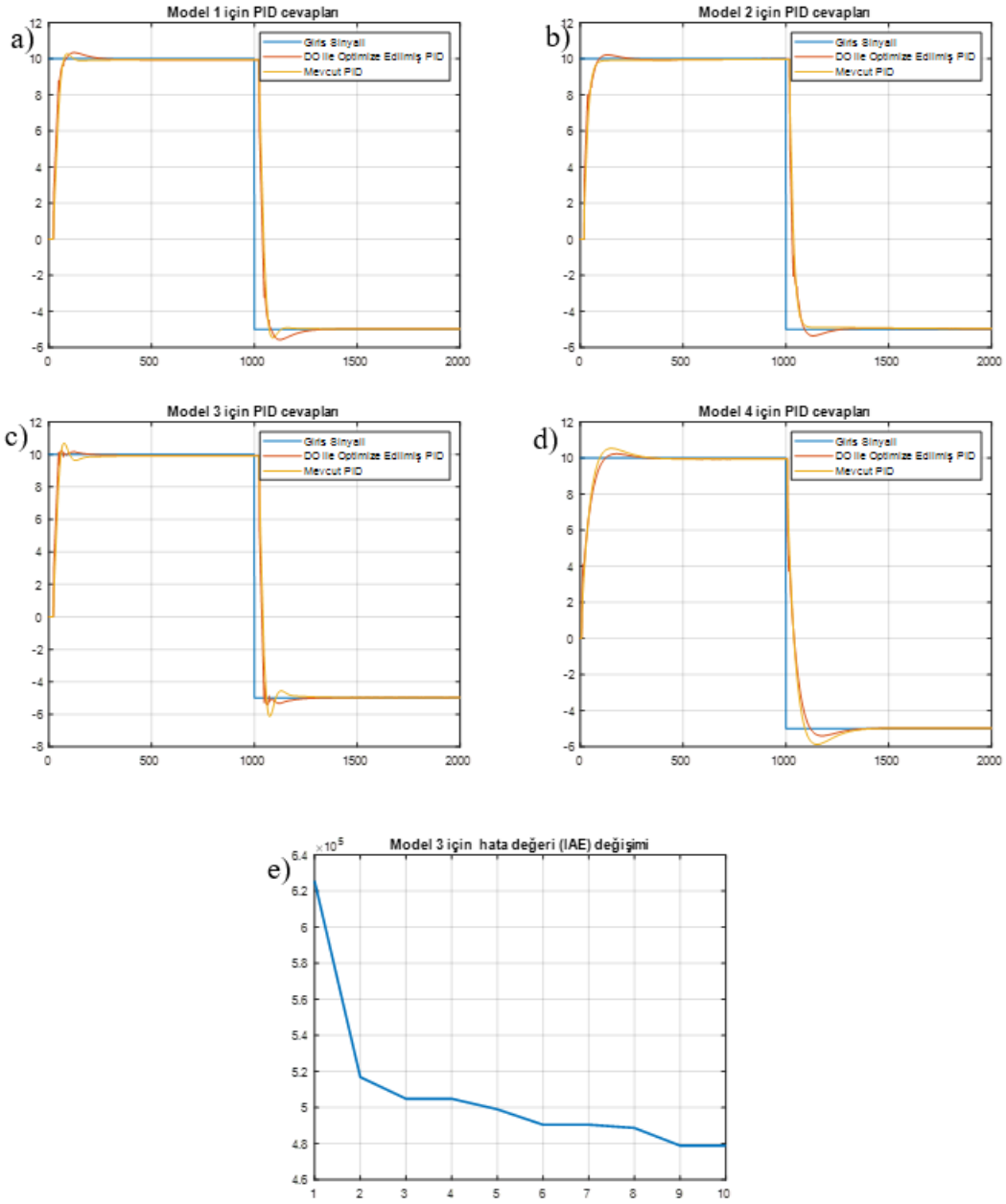
Şekil 3: Model 1 için Denge optimizasyon algoritması ile elde edilmiş PID kontrolörün giriş sinyali cevaplarının karşılaştırması.

Figure 3. *Comparison of the input signal responses of the PID controller obtained with Equilibrium optimization algorithm for Model 1.*



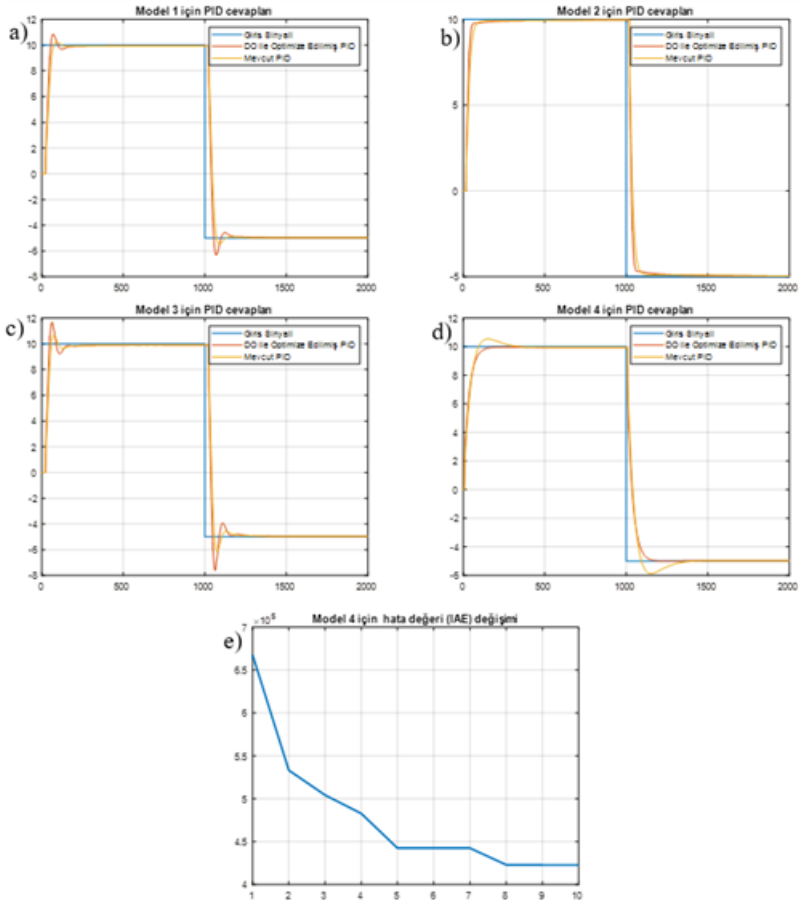
Şekil 4: Model 2 için Denge optimizasyon algoritması ile elde edilmiş PID kontrolörün giriş sinyali cevaplarının karşılaştırması.

Figure 4. *Comparison of the input signal responses of the PID controller obtained with Equilibrium optimization algorithm for Model 2.*



Şekil 5: Model 3 için Denge optimizasyon algoritması ile elde edilmiş PID kontrolörün giriş sinyali cevaplarının karşılaştırması.

Figure 5. *Comparison of the input signal responses of the PID controller obtained with Equilibrium optimization algorithm for Model 3.*



Şekil 6: Model 4 göre Denge optimizasyon algoritması ile elde edilmiş PID kontrolörün giriş sinyali cevaplarının karşılaştırması.

Figure 6. *Comparison of the input signal responses of the PID controller obtained with Equilibrium optimization algorithm according to Model 4.*

SONUÇ ve TARTIŞMALAR (RESULTS and DISCUSSIONS)

Bu çalışmada yeni bir meta sezgisel yöntem olan denge optimizasyon algoritması gerçek bir mühendislik probleminde kullanılmıştır. Aşı üretiminde önemli bir aşama olan karıştırıcı hızına karşın çözünen oksijen miktarının kontrolü için PID kontrolör parametreleri denge optimizasyon algoritması ile tasarlanmıştır. Öncelikle optimizasyon algoritması üç boyutlu bir optimizasyon problemi olan PID kontrolör parametrelerini optimize edebilecek kabiliyete ulaştırılmıştır.

İlgili sistem için literatürde önerilmiş dört farklı model için denge optimizasyon algoritması çalıştırılmış ve her model için elde edilen kontrolör parametreler diğer modeller içinde denenmiştir. Öncelikle denge optimizasyon algoritması referans alınan her model için mevcut durumda daha iyi kontrol performansı gösterebilen kontrolörler türetmiştir. Özellikle yükselme ve yerleşme zamanlarında üstünlükleri bulunmaktadır. Elde edilen sonuçlar denge optimizasyon algoritmasının mühendislik problemleri için özellikle kontrolör tasarımında kullanılabilecek uygun bir algoritma olduğunun göstergesidir.

Çalışma esnasında denge optimizasyon algoritması çok detaylı irdelenmiş ve özellikle algoritmanın çalışmasında son derece önemli olan bazı kat sayıların kontrolör tasarımı için hangi değerler alması gerektiği yapılan testler sonucunda tespit edilmiştir.

Gelecek çalışmaları olarak bu algoritmanın modifiye edilerek gerçek zamanlı sistem modelleme, yol belirleme vb. mühendislik problemlerinde kullanılması planlanmaktadır.

KAYNAKLAR (REFERENCES)

Atashpaz-Gargari, Esmaeil, and Caro Lucas. 2007. "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition." In *2007 IEEE Congress on Evolutionary Computation, CEC 2007*,.

Ateş, A., and C. Yeroglu. 2016. "Optimal Fractional Order PID Design via Tabu Search Based Algorithm." *ISA Transactions* 60.

Ateş, A., and C. Yeroğlu. 2018. "Modified Artificial Physics Optimization for Multi-Parameter Functions." *Iranian Journal of Science and Technology - Transactions of Electrical Engineering* 42(4).

Bazaraa, Mokhtar S., Hanif D. Sherali, and C. M. Shetty. 2005. Nonlinear Programming: Theory and Algorithms *Nonlinear Programming: Theory and Algorithms*.

Dorigo, Marco, and Krzysztof Socha. 2007. "Ant Colony Optimization." In *Handbook of Approximation Algorithms and Metaheuristics*,.

Faramarzi, Afshin, Mohammad Heidarinejad, Brent Stephens, and Seyedali Mirjalili. 2020. "Equilibrium Optimizer: A Novel Optimization Algorithm." *Knowledge-Based Systems*.

Gandomi, Amir Hossein, Xin She Yang, and Amir Hossein Alavi. 2013. "Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems." *Engineering with Computers*.

Guo, Zhishi. 2002. "Review of Indoor Emission Source Models. Part 1. Overview." *Environmental Pollution*.

Karaboga, Dervis, and Bahriye Akay. 2009. "A Comparative Study of Artificial Bee Colony Algorithm." *Applied Mathematics and Computation*.

Kaveh, A., and S. Talatahari. 2010. "A Novel Heuristic Optimization Method: Charged System Search." *Acta Mechanica*.

Khan, Omar et al. 2018. "Optimized PID Controller for an Industrial Biological Fermentation Process." *Journal of Process Control*.

Kuhn, Harold W., and Albert W. Tucker. 2014. "Nonlinear Programming." In *Traces and Emergence of Nonlinear Programming*,.

Lin, Ming Hua, Jung Fa Tsai, and Chian Son Yu. 2012. "A Review of Deterministic Optimization Methods in Engineering and Management." *Mathematical Problems in Engineering*.

Mirjalili, Seyedali et al. 2017. "Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems." *Advances in Engineering Software*.

Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. 2014. "Grey Wolf Optimizer." *Advances in Engineering Software*.

Rabani, Yuval. 2007. "Linear Programming." In *Handbook of Approximation Algorithms and Metaheuristics*,.

Rao, R. V., V. J. Savsani, and D. P. Vakharia. 2011. "Teaching-Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems." *CAD Computer Aided Design*.

Rashedi, Esmat, Hossein Nezamabadi-pour, and Saeid Saryazdi. 2009. "GSA: A Gravitational Search Algorithm." *Information Sciences*.

Yeroğlu, C., and A. Ateş. 2014. "A Stochastic Multi-Parameters Divergence Method for Online Auto-Tuning of Fractional Order PID Controllers." *Journal of the Franklin Institute* 351(5).

Zhang, Chunlei, and Raúl Ordóñez. 2012. "Numerical Optimization." In *Advances in Industrial Control*,.