

YSA VE GA TEMELLİ YENİ BİR ALGORİTMA İLE DOĞRUSAL OLMAYAN OPTİMİZASYON

Sabri Erdem*

ÖZET

Bu çalışmada doğrusal olmayan optimizasyon problemlerinin çözümünde yapay sinir ağlarının ve genetik algoritmaların kullanımıyla ilgili yeni bir yaklaşım önerilmektedir. Önerilen optimizasyon metodu, kısıtları öğrenmek için yapay sinir ağları, global optimum çözüme yakınsamak için genetik algoritmayı ve özellikle bazı kısıtların olurlu çözümü ihlal ettiği durumlarda metodun sonuçlarını değerlendirmek için ise amaç programlamayı seçenek çözüm olarak sunmaktadır. Yöntemin klasik yöntemlere göre hesaplama karmaşıklığı açısından avantajları incelenerek kullanım sınırlamaları belirlenmiştir.

Anahtar Sözcükler: *Doğrusal Olmayan Optimizasyon, Yapay Zekâ, Yapay Sinir Ağları, Genetik Algoritma, Hesaplama Karmaşıklığı*

OPTIMIZATION USING AN ANN and GA BASED NEW ALGORITHM

ABSTRACT

In this study, a new approach is offered to nonlinear optimization problem solving by using artificial neural networks and genetic algorithms. The proposed approach utilizes Artificial Neural Network (ANN) for learning constraints, Genetic Algorithm (GA) for evolving to global optimum solution and Goal Programming for evaluating results in case of some constraints violate their boundaries near around the feasible region. The approach is compared with classical methods in terms of computational complexity advantages and usage limitations are discussed.

Keywords: *Nonlinear Optimization, Artificial Intelligence, Artificial Neural Network, Genetic Algorithm, Complexity*

* Dokuz Eylül Üniversitesi, İşletme Fakültesi, Tınaztepe Kampüsü Buca, İzmir, E-posta: sabri.erdem@deu.edu.tr

GİRİŞ

İşletme, iktisat, bu bilim dallarının türevleri, mühendislik alanları ve diğer disiplinler, özellikle daha az girdi kullanımıyla daha çok çıktı elde etme düşüncesiyle her zaman en iyiyi arama çabası içerisindeyler. Söz konusu çabalar bilim adamlarını ve araştırmacıları, düşüncelerini analitik modellere aktarmak yoluyla gerçek hayat problemleri için en iyi çözümü tanımlamaya yöneltmektedir. Bu modeller arasında gerçek hayat problemlerini en büyükleme ve en küçüklemeyle dayandıran modeller optimizasyon modelleri olarak değerlendirilebilirler. Gerçek hayat problemlerinin optimizasyon modelleri, problemlerin doğası gereği kaynak kısıtları, çalışma alt limitlerinin sağlanması gibi genellikle çok sayıda kısıt içermektedir.

İşletme problemlerinin ifade edilebilmesi için sıklıkla doğrusal programlama modelleri kullanılmakla birlikte, doğrusal olmayan optimizasyon yaklaşımları ise doğrusal olmayan ilişkilerin incelenebilmesi için için kullanılmaktadır. Bu yaklaşımlar arasında en sık başvurulanlar, tamsayı programlama ve karesel (quadratic) programlama modelleridir. Diğer taraftan karar değişkenleri arası ilişkilerin daha yüksek dereceli polinom, trigonometrik ya da logaritmik vb. olması durumunda; kesikli ve sürekli fonksiyonlar olması durumunda ve olurlu çözüm bölgesinin konveks olması ya da olmaması durumunda, optimizasyon modellerine farklı çözüm yaklaşımları da getirilebilmektedir. Bu yöntemlerden sürekli ve türevli fonksiyonlar için, en yaygın olanı, Lagrange ve Genelleştirilmiş İndirgenmiş Gradyen yöntemidir. Türevli olmayan ya da konveks olmayan olurlu çözüm bölgesine sahip problemler için kullanılan yöntemler üzerinde halen yeni çalışmalar yapılmakta ve yeni yaklaşımlar geliştirilmeye çalışılmaktadır.

Son dönem gerçekleştirilen çalışmalarda kullanılan yaklaşımlar arasında sezgisel yaklaşımlar oldukça önemli bir paya sahiptir. Çözüm alanının kesikli yapıda olması ya da kısıt fonksiyonlarının türevli olmaması durumunda optimalite testi yapmak da imkansızlaşmaktadır. Özellikle geliştirilen ortak test problemlerinde, "şimdiye kadar bulunan en iyi" yaklaşımı getirilerek araştırmacıların kendi yöntemlerinin performansını bu sayede karşılaştırma olanakları yaratılmıştır. Bu çerçevede, en iyi çözümün ne olduğu kadar, sonuca hızlı ulaşabilmek de önemlidir. Dolayısıyla geliştirilen modellerin hesaplama karmaşıklığı da bu tür karşılaştırmalarda dikkatle izlenmektedir.

Bu çalışmada hem konveks hem de konveks olmayan olurlu çözüm bölgelerine sahip doğrusal olmayan optimizasyon problemleri için türevli olma koşulu gerektirmeyen yeni bir algoritma önerilmektedir. Bugüne

YSA ve GA Temelli Yeni Bir Algoritma İle Doğrusal Olmayan Optimizasyon

kadar geliştirilen yaklaşımlar genelde türev bağımsız (Rodrigueza vd., 2000), kısıtsız optimizasyon problemleri için direkt arama (Lewis vd., 2000), kısıtlı optimizasyon problemleri için birçok meta-sezgisel yöntem ve genetik algoritmalar (Wah ve Chen, 2001), büyük ölçekli problemleri inceleyen karşılaştırmalı çalışmalar (Benson vd., 2001) ve iteratif yöntemler (Kelley, 1999) olarak özetlenebilir.

Çalışmanın ilk bölümünde doğrusal olmayan modellerin genel yapısı, Yapay Sinir Ağları (YSA) ve Genetik Algoritmalar (GA) hakkında tanıtıcı bilgiler verilmekte, ikinci bölümünde, bu çalışmada ele alınan yaklaşım sunulmakta ve son bölümde ise yöntemin kullanılabilirliği ve katkıları tartışılmaktadır.

MATERYAL

Kısıtlı doğrusal olmayan problemlerin genel yapısı Denklem 1'deki gibi verilebilir.

$$\min f(x)$$

kısıtlar :

$$g_i(x) \leq b_i \quad i = 1, 2, \dots, m \quad (1)$$

$$h_j(x) = k_j \quad j = m + 1, \dots, n$$

$$\forall x_i \& \forall x_j \geq 0$$

Burada m adet $g(x)$ fonksiyonu eşitsizlik kısıtlarını, n adet $h(x)$ fonksiyonu eşitlik içeren kısıtları, b ve k değişkenleri eşitsizlik ve eşitlik kısıtlarının sağ taraf değerlerini ve $f(x)$ ise amaç fonksiyonunu temsil etmektedir. Ayrıca gerçek hayat problemlerine özgü olarak karar değişkenlerinin negatif olmama koşulu da kısıtlara eklenmiştir. Doğrusal olmama genellemesinden dolayı, modeldeki bütün fonksiyonlar doğrusal olmayan içeriğe sahip olabilirler.

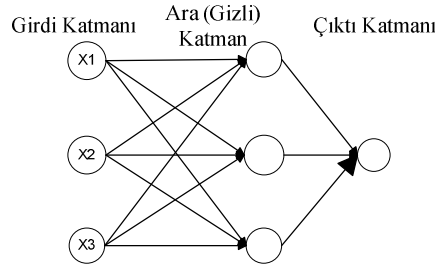
Denklem 1 ile verilen optimizasyon modelleri için son dönemlere kadar Newton matematiğini temel alan gradyan ve türev alma eksenli çözüm yaklaşımları kullanılmıştır. Hesaplama bilimindeki gelişmeler ve bu gelişmelerin gerçekleştirilebileceği teknoloji ve bilgisayar donanımı alt yapısı sayesinde, 1 saniye içerisinde milyarlarca işlem yapabilme hızından yararlanabilen ve sezgisel yaklaşımlar içeren çözüm yöntemleri geliştirilmiştir. Bu kapsamda, özellikle son yıllarda kısıtlı doğrusal olmayan

Sabri Erdem

optimizasyon ile ilgili problemlerin çözümünde YSA ve GA yaklaşımları da kullanılmaya başlanmıştır.

Yapay Sinir Ağları

Biyolojik sinir hücrelerinin fonksiyonel özelliklerinden ilham alınarak McCulloh ve Pitts (1943) tarafından ilk olarak ortaya atılan yapay sinir ağları günümüzde tahminleme, sınıflandırma, kümeleme ve tanıma gibi farklı alanlarda kullanılmaktadır. YSA'lar kullanım amaçlarına göre farklı yapılara sahip olmaktadır. Bu çalışmada kullanılan YSA yöntemi ileri beslemeli ve geriye hata yayımlı çok katmanlı algılayıcı olarak tanımlanmaktadır. Şekil 1 incelendiğinde girdi, çıktı ve gizli katman olmak üzere üç katmana sahip, 3 girdi (x_1 , x_2 ve x_3), tek çıktı ve tek gizli katman ve bu katmanda 3 gizli nörona sahip bir basit bir YSA yapısı görülmektedir.



Şekil 1: Basit Çok Katmanlı Algılayıcı YSA Modeli

YSA modellerinde esas olan veri setini öğrenme ve öğrenme süreci içerisinde ortaya çıkan hatayı ağ üzerinde yayma biçimidir. YSA modelleri, kendilerine sunulan test setindeki girdi çıktı ilişkilerine sahip olduğu düşünülen verileri, kara kutu yapısı içinde açıklayabilecek biçimde, nöronlar arasındaki bağlantı ağırlıklarını değiştirmek suretiyle öğrenirler. Öğrenme süreci, eğitim setindeki her bir girdi değerinin ağa sunulması ve buna karşılık gelen çıktı değeri ile gerçek çıktı değeri arasındaki farkın ağ üzerinde geriye doğru yayılarak ağıdaki nöronlar arası bağlantı ağırlıklarını değiştirilmesi suretiyle olmaktadır.

YSA'nın eğitim setindeki bütün vektörler için ürettiği çıktı ile gerçek çıktı arasındaki farkı gösteren hata değerinin önceden belirlenen bir limit değerinin altında olması durumu YSA'nın veriler arası ilişkiyi öğrenmesi olarak değerlendirilir. Bu çalışmada kullanımı bakımından ise, kısıtlardan en az birinin sağlanamaması durumunda YSA'nın o çözüm vektörü için "kısıtları sağlamıyor" ve bütün kısıtların sağlanması durumunda ise

YSA ve GA Temelli Yeni Bir Algoritma İle Doğrusal Olmayan Optimizasyon

“kısıtları sağlıyor” cevaplarına karşılık gelebilecek sayısal değerleri (sırasıyla 0 ve 1 değerleri) belirli bir hata payının altında üretmesi, kurulan YSA modelinin kısıtları öğrenmeyi tamamladığı anlamına gelir.

Burada YSA'nın; amaç fonksiyonu, eşitlik ve eşitsizlik içerebilen kısıtları ile sınır koşulları bilinen optimizasyon probleminin kısıtlarını, Denklem 1'de modellendiği gibi, problemin büyüklüğüne de bağlı olarak (yani karar değişkeni sayısı, kısıt sayısı, içerdiği doğrusal olmayan fonksiyonların yapısı vb.) kabul edilebilir bir hata düzeyinde öğrenmesi hedeflenmiştir. Bu kısıtları öğrenmesi beklenen YSA genel yapısı da Şekil 1'de verilmektedir. Buna göre oluşturulacak olan YSA'nın çalışma parametreleri, gizli katmanı ve gizli katmandaki nöron sayısı da optimizasyon problemi için belirlenmektedir.

Genetik Algoritmalar

Holland (1975) tarafından ortaya atılan GA kavramı, canlı nesilleri içindeki çeşitlilik, bu çeşitliliği yaratan mutasyon ve çaprazlama gibi genetik işlemleri konu almaktadır. GA, esil özelliklerinin daha iyi bireyler oluşturabilmek amacıyla sonraki nesle aktarılması gibi temel süreçleri, bir gerçek hayat probleminin çözümünde kullanılmak üzere evrimsel çözüm yöntemlerine adapte edilmesidir. Günümüzde GA özellikle çok fazla disk ve bellek ihtiyacı olan optimizasyon problemlerinde tercih edilmektedir.

GA değerlendirdiği nesil içindeki bireylerin uygunluk fonksiyonlarına (amaç fonksiyonu) göz önüne alarak, ortalamanın ya da önceden belirlenen başka bir eşik değerinin altında olanları ortadan kaldırmakta ve yerine bir sonraki iterasyonda değerlendirilmek üzere yeni bireyler (çözüm noktaları) yaratarak YSA'ya kısıt sağlama testi için göndermektedir.

Yeni nesilde de yaşamını devam ettiren bireyler son çözüme ulaşmak için yeni bireylerin yaratılmasında kilit rol oynayan aile bireyleri olarak değerlendirilirler. Yeni bireylerin neslin evriminde daha etkili oldukları düşünülmektedir. Özellikle GA mutasyon operatörü çeşitliliğin artmasında kilit rol oynamakta ve böylece yerel optimumdan kaçınılabilecek bir mekanizma sağlanmış olur.

GA bir nesil içerisindeki en iyi bireyi, sahip olduğu elitizm stratejisi yardımıyla bir sonraki nesle hiçbir değişikliğe uğratmadan geçirebilmektedir. Bunun yanı sıra GA modeli tasarlanırken tek noktadan ya da çoklu noktadan çaprazlama, doğrusal ya da karesel çaprazlama fonksiyonu (Haupt and Haupt, 2004), tek ya da çok noktalı mutasyon (Melanie, 1999), ekstrapolasyon stratejisi fonksiyonu ve rastgele sayı

Sabri Erdem

üretimi seçimleri de dikkate alınmalıdır. Bu seçimlerinin her birinin sonuç üzerinde farklı etkileri olmakla birlikte GA ile ilgili bazı yaşamsal soruların cevaplanması beklenmektedir:

- Hangi stratejiler kombinasyonu daha iyidir?
- Seçilen bu kombinasyon bütün kısıtlı optimizasyon problemlerinde uygulanabilir mi?
- Sonucun doğruluğu nasıl onaylanabilir?

Buradaki her bir soru detaylı araştırma çabalarını gerektirir. Tüm bu soruların cevaplarını aramak bu çalışmanın kapsamı dışındadır. Ancak çalışma kapsamından fazla çıkmamak kaydı ile konuyla ilgili bazı çalışmalara yer verilmeye çalışılmıştır. Şüphesiz ki her bir problemin kendine özgü özellikleri bulunmaktadır. Ancak çözüm teknikleri, sezgisel yaklaşımlar kullanıldığında, akıllı çözüm algoritmaları içerebilmektedir. Bu noktada ise çözüm yaklaşımı ne kadar sezgisellik içeriyorsa daha az parametre ile dahi daha iyi sonuç elde edilir.

YÖNTEM

YSA ve GA'nın Doğrusal Olmayan Optimizasyon'da Kullanımı

Bu çalışmada GA kısıtsız ve sınırlı optimizasyon problemlerinde uygulandığı biçimiyle en iyi çözümü bulma aracı olarak kullanılmaktadır. YSA'dan ise GA'nın kısıtsız bir optimizasyon problemiyle çalışabilmesi için kısıtları değerlendirici bir ara araç olarak yararlanılmaktadır. Genel bir ifade ile yöntemin çalışma biçimi; değişkenlerin sınır koşulları arasında kalan arama bölgesinden elde edilen rastgele çözüm noktalarının, kısıtları sağlama ve sağlayamama durumlarının YSA tarafından öğrenilmesi ve buna bağlı olarak hesaplanan amaç fonksiyonu değerinin GA ile optimize edilmek üzere değerlendirilmesi, yeni nesiller oluşturulması ve bu nesillerdeki bireylerin de tekrarlı biçimde YSA ile başlayan çevrime girmesi üzerine kuruludur. Detaylı işleyiş Tablo 1'de verilmektedir.

YSA modeli ile gerçek çıktı arasındaki ata seviyesi ile ilgili bazı görüşler öne sürülmektedir. Bunlardan ilki problemin karmaşıklığına ve optimizasyon probleminin doğasına bağlı olarak farklı hata düzeylerinin olabileceğidir. İkincisi ise, yüksek hata düzeylerinin yüksek olasılıklara karşın tercih edilmeyeceğidir. Sonucu olarak yüksek hata düzeyi altında bazı hata düzeltme ve düzgunleştirme mekanizmaları varlığıdır. Bu durum çoğu zaman genellikle doğrusal dönüşümleri temel alan veri normalizasyonu teknikleriyle başarılmaktadır.

YSA ve GA Temelli Yeni Bir Algoritma İle Doğrusal Olmayan Optimizasyon

Tablo 1: Önerilen Yöntemin Detaylı İşleyişi

<ol style="list-style-type: none">1. Problemin kurulması,<ol style="list-style-type: none">a. Karar değişkenlerinin ve sınırlamaların tanımlanması,b. Kısıtların tanımlanması,c. Amaç fonksiyonunun tanımlanması,2. Karar değişkenlerinin sınır değerleri arasında başlangıç çözüm kümesinin tanımlanması,<ol style="list-style-type: none">a. YSA Öğrenmesi için rastgele bir alt çözüm kümesi belirlenmesi,<ol style="list-style-type: none">i. Her bir çözüm vektörünün bütün kısıtları sağladığının test edilmesi,3. YSA'nın çözüm kümesini öğrenmesi,<ol style="list-style-type: none">a. Öğrenme ve geriye yayılım oranının belirlenmesi,b. Katmanlar ve ara nöron sayısının belirlenmesi,4. YSA'nın testi için performans kümesinin seçilmesi,<ol style="list-style-type: none">a. Başlangıç çözüm kümesi içinde öğrenme için kullanılmayanların belirlenmesi,b. YSA modelinin test edilmesi,5. YSA'nın öğrenip öğrenemediğine karar verilmesi,6. Olurlu çözümlerin GA'ya iletilmesi,7. GA sezgisel/uygunluk fonksiyonunun belirlenmesi,8. GA parametrelerinin/politikalarının belirlenmesi,<ol style="list-style-type: none">a. Çaprazlama oranı ve türüne karar verilmesi (i.e., tek ve çok noktalı, doğrusal ve karesel),b. Mutasyon oranı ve türü,c. Elitizm,d. Öldürme yöntemi,e. Yeni neslin yaratılması,9. Her bir bireyin uygunluk değerlerinin hesaplanması,10. Bireyleri yaşatma ve öldürmenin belirlenmesi,11. Bulunan en iyi çözümün belirlenmesi,<ol style="list-style-type: none">a. Karar politikaları ve kriterleri, <p>SONRAKİ NESİL</p> <ol style="list-style-type: none">12. Bir önceki nesilde ölen bireylerin yerine yenilerinin yaratılması,<ol style="list-style-type: none">a. Çaprazlama uygulanması,b. Mutasyon uygulaması,13. Yeni neslin ikinci iterasyon olarak GA'da kullanılacak çözümlerin bulunup YSA'ya gönderilmesi,14. Çözüm süreci boyunca YSA için öğrenme sonrası politikaların yaratılması,15. Durma koşulu sağlanana kadar 6 ve 14 arası aşamaların tekrar edilmesi.

Sabri Erdem

YSA'nın çıktı tarafında elde edilen değerleri denormalize etme aşamasında bazı sezgisel yaklaşımlar kullanılabilir. Bu ters dönüşüm sürecinde, 0 ile 1 arasında (genellikle 0.5 olarak seçilen) bir kesme ya da eşik değerini baz alan basit bir açık dönüşüm gerçekleştirilmektedir. Elde edilen çıktı değeri bu eşik değerinin altındaysa 0, eşit ve üzerindeyse 1 olarak alınmaktadır. Buna göre elde edilen çıktı değerinin 1 olması durumu bütün kısıtların sağlanma durumunu göstermektedir. Bu dönüşüm yöntemin daha kabul edilebilir bir hata düzeyine erişmesine yardımcı olmaktadır.

Tablo 1'deki 3. aşamaya göre, YSA'nın eğitim aşamasından sonraki aşamada amaç fonksiyonu değeri hesaplanarak, fonksiyon değerini daha iyi bir çözüme götürebilecek iterasyon adımları gerçekleştirilir. Bu iyileştirme adımları ise GA yöntemi aracılığıyla gerçekleştirilebilmektedir.

Son aşama GA'nın bulgularının değerlendirilmesidir. YSA'nın öğrenme aşamasında göreceli olarak yüksek hata oranı ile karşılaşılabilir. Modelin GA ile çözümü aşamasında, kısıtların çok büyük bir bölümü sağlanmaktadır. Diğer bir deyişle, YSA'nın öğrenme aşamasındaki hata oranının küçük olması daha iyi kısıt sağlama oranının elde edilmesini sağlayacaktır. Yine de, eğer çözüm optimuma yakın ise kısıtların büyük çoğunluğu sağlanır ve sağlanmayan kısıtlar üzerinde ayrıca bir araştırma gerçekleştirilebilir.

Önerilen Yöntemin Hesaplama Karmaşıklığı

Büyük ölçekli optimizasyon problemleri çok sayıda değişken ve doğrusal olmayan kısıtlara sahiptir. Çözücüler optimal çözümü bulmak için çalıştırıldığında, klasik literatürde, her bir iterasyonda bütün kısıtları sağlayıp sağlamadıklarını test etmektedirler. Ya da en azından ceza fonksiyonu olarak amaç fonksiyonu içine gömülü olarak yine her bir iterasyonda tekrar hesaplanmaktadır. Bu çalışmanın ilgili literatüre önemli katkılarından birisi, yüzlerce kısıtla sahip bir problemin çözüm sürecindeki her iterasyonda bütün kısıtları tekrar test etmek yerine sadece bir YSA kara kutusunda daha basit bir hesaplama ile sonuca ulaşabilmesidir.. YSA modelinin hesaplama karmaşıklığı 2 numaralı gösterimdeki gibidir.

YSA ve GA Temelli Yeni Bir Algoritma İle Doğrusal Olmayan Optimizasyon

$$(n_i + 1)n_h + (n_h + 1)n_o \Rightarrow O(n_i n_h)$$

burada

$$n_i = \text{girdi sayısı}, \quad (2)$$

$$n_h = \text{ara katman nöron sayısı},$$

$$n_o = \text{çıkıktı nöron sayısı}.$$

Klasik yöntemler kısıtları işlerken 3 numaralı gösterimdeki gibi bir hesaplama karmaşıklığına sahiptirler.

$$n_c n_{avgterm}$$

burada

$$n_c = \text{kısıt sayısı}, \quad (3)$$

$$n_{avgterm} = \text{ortalama hesaplanan terim sayısı}$$

Diğer taraftan, problemin karmaşıklığına da bağlı olarak, klasik yöntemlerin arama zamanı ile önerilen yöntemin öğrenme zaman gereksinimi arasında bir ikame söz konusudur. Bu çalışmada önerilen yöntem, etkin çözüm zamanı açısından daha uygun olacağından karmaşıklığı yüksek olan problemlerde uygulanmalıdır. Hangi çözüm yöntemin seçilmesi gerektiğine arama zamanı açısından karar verebilmek için kullanılacak karşılaştırma 4 Numaralı gösterimdeki gibi açıklanabilir.

$$t_{train} + t(n_i + 1)n_h + (n_h + 1)n_o n_{avg_itr_ga} \leq n_c n_{avgterm} n_{itr_trad_meth} \quad (4)$$

burada

$$t_{train} = \text{YSA eğitim zamanı},$$

$$t = \text{her bir iterasyon zamanı},$$

$$n_{avg_itr_ga} = \text{ortalama GA iterasyon sayısı},$$

$$n_{avgterm} = \text{ortalama hesaplanan terim sayısı},$$

$$n_{itr_trad_meth} = \text{klasik yöntem iterasyon sayısı}.$$

3 ve 4 numaralı gösterime dayanarak, YSA eğitim sürecinin zaman karmaşıklığı, deneme sayısı (Epoch), eğitim kümesi, girdi sayısı ve gizli nöron sayısı dikkate alındığında, 5 Numaralı denklemdeki gibi formüle edilebilir.

Sabri Erdem

$$t_{train} \approx Epoch(11+(n_i+2)n_h+n_p(16+n_h(6n_i+4)))t_u$$
$$\Rightarrow O(n_{epoch} n_p n_h n_i) \quad (5)$$

burada

n_p = eğitim kümesi eleman sayısı,

t_u = her bir aritmetik mantuk işlemi için

gereken ortalama hesaplama zamanı

GA'nın iterasyonlarının ortalama zaman karmaşıklığı Denklem 6'da verilmektedir.

$$t_{ga_iterations} = 10t_u + t_{build_pop} + t_{evolve}$$

burada (6)

t_{build_pop} = ilk nesli oluşturma zamanı,

t_{evolve} = birçok kez yeni nesil yaratmak için gerekli zaman.

$$t_{build_pop} = 14t_u + n_{ind}(2t_u + n_i t_u)$$

burada (7)

n_{ind} = popülasyondaki birey sayısı.

Big_OH kuralı (Zanden, 2008) temel alındığında GA'nın karmaşıklığı 8 numaralı eşitlikteki gibidir.

$$T(n)_{ga_iterations} = T(n)_{evolve}$$
$$T(n)_{ga_iterations} = O(n_{ind}(n_{exp_rnd} + n_h n_i)) \quad (8)$$

Böylece 4 numaralı gösterimdeki eşitsizliğin sol tarafındaki karmaşıklık Denklem 9'daki gibi olmaktadır.

$$T(n) = O(n_{epoch} n_p n_h n_i + n_{ind}(n_{exp_rnd} + n_h n_i)) \quad (9)$$

Bununla birlikte, Denklem 4'teki eşitsizliğin sağ tarafındaki zaman karmaşıklığı n değişken ve k iterasyon için kabaca $T(n) = O(kn^2)$ olarak tanımlanabilir.

YSA ve GA Temelli Yeni Bir Algoritma İle Doğrusal Olmayan Optimizasyon

$$O(n_{epoch}n_p n_h n_i + n_{ind}(n_{exp_rnd} + n_h n_i)) \leq O(kn^2) \quad (10)$$

t_u , soyut zaman olarak da bilinen (Zanden, 2008), ortalama olarak basit bir işlemi gerçekleştirme zamanını göstermektedir. Bu çalışmada t_u soyut zamanının büyüklüğü deneysel olarak bulunmuştur. YSA'nın eğitim süresince belirlenen deneme sayısı ve ara nöron sayısı arttıkça t_u birim işlem zamanı belirli bir değere yaklaşmaktadır. Bu t_u birim işleme zamanları 10^{-8} saniye cinsinden Tablo 2'de verilmektedir.

Tablo 2: Birim İşlem Zamanı (10^{-8} s)

		Ara nöron sayısı	
		5	10
Deneme Sayısı	1000	8.24	7.69
	5000	7.40	7.46
	10000	7.08	7.29
	20000	7.16	7.24

SONUÇ VE ÖNERİLER

Optimizasyon modelleri gerçek hayat problemleri üzerinde akılcı kararlar alabilmek için kullanılan yardımcı araçlardır. Bütün durumların sonuçları karar destek sistemi için bir girdidir ve önceden belirlenmiş olan kısıtları temel alırlar. Önceden belirleme sürecinden farklı olarak bazı kısıtlar, zamanla ve problemin doğasıyla değişebilen farklı karakteristiklere sahiptirler. Bu çerçevede düşük oranda dahi olsa sağlanmayan kısıtların gerçek dünyadaki anlamları ve sağlanmama oranları tekrar gözden geçirilip araştırılabilir.

Modelin işletilmesi sonucunda sağlanamayan kısıtların bulunduğu son çözümlerde kısıtlar sıkı ve esnek kısıtlar olarak ayrılabilir. Eğer bir ya da daha fazla kısıt esnek kısıtlara dönüştürülebilirse optimizasyon problemi hedef programlama problemine dönüşebilmektedir. Bu durum var olan çözüm yöntemlerine başka bir kapı açmaktadır. Hatta bu sayede sağlanmayan kritik kısıtların sağ taraf değerleri üzerinde çeşitli duyarlılık ve senaryo analizleri gerçekleştirmek de mümkün olabilir. Böylece duyarlılık analizleri ve problem kısıtlarının gerçek dünya anlamları arasında bir bağlantı kurulmuşması mümkün olabilir.

Hedef programlama modelleri hem sıkı hem de esnek kısıtlara sahiptirler. Sıkı kısıtların sağlanması zorunlu olmakla birlikte esnek

Sabri Erdem

kısıtlargenelliklebelirli önceliklere göre belirli sapmalar ile sağlanırlar. Eğer mümkünse sıkı kısıtların esnek kısıtlara dönüştürülebilmesi amaç fonksiyonlarında önemli değişikliklere ve çözüm bulmada esnekliklere yol açabilir. Burada kısıtların sağlanmama oranı ile kısıtları da içeren amaç fonksiyonların değeri arasında güçlü bir ilişki bulunmaktadır. Bu durum özellikle sağlanmayan kısıtların oransal olarak arttığı ve amaç fonksiyonunun hızlı biçimde iyileştiği durumlar için daha önemlidir.

Çözüm süreciyle ilgili bazı görüşler olabilir. Birincisi, bu çalışmadaki modelden farklı olarak, YSA'nın daha iyi öğrenme oranları elde edebilmesi için GA'nın kullanılmasıdır. GA kullanılarak YSA'da başarılı öğrenme sonuçları elde edilmiştir (Kitano, 1994). İkinci olarak, YSA'da nöronlar arası bağlantıların ağırlıkları GA aşamasının sonuçları üzerinde, popülasyondaki evrimleşmenin yönü bakımından değişikliklere sebep olabilir. Başka bir deyişle farklı denemelerde farklı sonuçlar elde edilebilir.

KAYNAKÇA

- Benson, H.Y., Shanno, D.F. & Vanderbei, R.J. (2001). A Comparative Study of Large-Scale Nonlinear Optimization Algorithms, Proceedings of the Workshop on High Performance Algorithms and Software for Nonlinear Optimization, Erice, Italy.
- Haupt, R.L. & Haupt, S.E. (2004). Practical Genetic Algorithm 2nd Edt., John Wiley and Sons Inc. Publication, New Jersey.
- Holland, J.H. (1975). Adaptation in Natural and Artificial Systems, An Harbor, University of Michigan Press.
- Kelley, C.T. (1999). Iterative Methods for Optimization, Published by SIAM
- Kitano, H. (1994). Neurogenetic Learning: An Integrated Method of Designing and Training Neural Networks Using Genetic Algorithms, Physica D, 75: 225–238.
- Lewis, R.M., Torczon, V. & Trosset, M.W. (2000). Direct Search Methods: Then and Now. Journal of Computational and Applied Mathematics, 124: 191-207.
- Mitchell, M. (1999). An Introduction to Genetic Algorithms 5th Edt., London.
- McCulloch, W.S. & Pitts, W.H. (1943). A Logical Calculus of The Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 5: 115-133.
- Rodrigueza, J.F., Renaudb, J.E., Wujekc, B.A. & Tappeta, R.V. (2000). Trust region model management in multidisciplinary design optimization, Journal of Computational and Applied Mathematics, 124: 139-154.

YSA ve GA Temelli Yeni Bir Algoritma İle Doğrusal Olmayan Optimizasyon

- Wah, B.W. & Chen, Y. (2001). Hybrid Constrained Simulated Annealing and Genetic Algorithms for Nonlinear Constrained Optimization, Proc. IEEE Congress on Evolutionary Computation, 925-932.
- Yu, N. (2008). Numerical Methods for Semidefinite Programming, İndirilme Tarihi: 1/3/2008. WWW:Web:[http://www.core.ucl.ac.be/Doctoral Courses /Lecture1.pdf](http://www.core.ucl.ac.be/Doctoral%20Courses/Lecture1.pdf)
- Zanden, B.V. (2008). Analysis of Algorithms and Selection of Algorithms, İndirilme Tarihi: 1/3/2008. WWW:Web:<http://www.cs.utk.edu/~bvz/bvz/classes/cs302/notes/complexity.html>