

[1245]

EVİRİMSEL ARAMA ALGORİTMALARIN KAOTİK DOĞASI VE GEOMATİK MÜHENDİSLİĞİNDE KULLANIM OLANAKLARI

Pınar ÇİVİCİOĞLU¹, M. Akif GÜNEN², A. Emin KARKINLI², Erkan BEŞDOK³

¹ Prof. Dr. Erciyes Üniversitesi, Uçak Elektrik Elektronik Bölümü, 38039, Kayseri, civici@erciyes.edu.tr

² Araş. Gör. Erciyes Üniversitesi, Harita Mühendisliği Bölümü, 38039, Kayseri, (akif:akarkinli)@erciyes.edu.tr

³ Prof. Dr. Erciyes Üniversitesi, Harita Mühendisliği Bölümü, 38039, Kayseri, ebesdok@erciyes.edu.tr

ÖZET

Evrimsel Arama Algoritmaları (EA), farklı problem tiplerine kolay uyarlanabilmeleri, esnek yapıları, amaca uygun tasarıma olanak sağlamaları ve yüksek problem çözüme kapasiteleri nedeniyle sayısal problemlerin çözümünde yaygın olarak kullanılmaktadır. Yaygın kullanımlarına rağmen EA'ların neden ilgili problemin çözümüne yakınsadıklarını gösteren analitik olarak modellenen bir açıklama bulunmamaktadır. Bu bildiride EA'ların kaotik doğaları, çalışma stratejileri ve temel tasarım ilkeleri üzerinde kısaca durulmuştur. Ayrıca EA'ların kümeleme, kamera kalibrasyonu, görüntü geliştirme ve koordinat dönüşümü gibi çeşitli problemlere nasıl adapte edilebildikleri kısaca gösterilmiştir.

Anahtar Sözcükler: Evrimsel Algoritmalar, Rassal Arama, Geri İzleme Algoritması.

ABSTRACT

CHOTIC NATURE OF EVOLUTIONARY SEARCH ALGORITHMS

Evolutionary Search Algorithms (EA) are extensively used for solving numerical problems because they can be easily adopted to different types of problems, they have flexible structures, they enable designs appropriate for the purposes and they have high problem solving capabilities. Although EAs have been frequently used in the literature, there is not an analitically modelled clarification pointing out why they converge to the solution of the related problem. In this paper, chaotic natures of EAs, their operational strategies and fundamental design principles have been examined. Furthermore, it is briefly shown how the EAs can be adapted to various types of problems such as clustering, camera calibration, image enhancement and transformation of coordinates.

Keywords: Evolutionary Search Algorithms, Random Search, Backtracking Search Algorithm.

1.EVRİMSEL HESAPLAMA VE KAOTİK DAVRANIŞ

Evrimsel Arama Algoritmaları (EA), çoklu-çözümlü, lineer olmayan formda veya differansiyeli tanımlanamayan, süreksiz, kısıtlı veya kısıtsız, kompleks veya katışık formda tanımlanan sayısal optimizasyon problemlerinin en iyilenmiş alt çözümlerinden en az birisine yakınsayabilen rastgele arama araçlarıdır (Çivicioğlu, 2013a,2013b,2013c, 2013d). EA'lar, çoğunlukla gradiyente dayalı klasik gürbüz -robust- en-iyileme algoritmalarına kıyasla belirgin avantaj ve dezavantajlara sahiptir; EA'lar ilgili probleme -genellikle- kolay uyarlanabilir, problemin türevlenmiş formlarına ihtiyaç duymaz, ilgili probleme ait yeterince iyi bir başlangıç çözümü de kullanmazlar. Buna karşılık EA'lar genellikle kabul edilebilir bir sonuca yakınsamamak için daha fazla iterasyona veya sınırlanmış arama uzayına ihtiyaç duyabilirler. Ayrıca EA'lar, problemin çoklu-çözümlü (multi-modal), tekli-çözümlü (uni-modal), komplekslik derecesi, problemin doğal parametre sayısı, algoritmanın kontrol parametrelerinin sayısı ve bunların başlangıç değeri gibi bir çok parametrenin başlangıçta yeterince iyi kestirilmiş olmasına da ihtiyaç duyarlar. Beklenmedik şekilde bir çok problem çözümünde güvenilir bir ölçüt olan MSE, MAE gibi hata ölçütlerine dayalı amaç fonksiyonları kullanmakta dış bir faktör olarak arama performansını azaltabilmektedir. Karşı karşıya kalınan problemin yapısal doğası EA'ların yerel çözümlerden kaçabilme yeteneğini doğrudan etkiler; problem tekli-çözüme sahipse öncelikle ön-seçici -elitist- bir yaklaşımla tasarlanmış bir algoritma test edilmelidir, problem çoklu-çözümlü bir yapıdaysa problemin yapısına göre başlangıçta kullanılacak rassal çözüm kümesini oluşturan pattern matrisinin boyutunu yeterince iyi kestirmek gerekebilir. Pattern matrisi boyutu gereğinden çok küçük veya çok büyükse problem çözümünde istenilen çözünürlüğe kolayca veya hiç erişilemeyebilir.

Literatürde tanıtılan çok sayıda EA bulunmaktadır; PSO, DE, Ant, CoBiDe, CoDE, ABC, BSA, ACS gibi. Sıradan bir EA pek çok klasik test problemini kolayca çözebilir. Buna karşın karmaşık test problemini çözebilen EA sayısı oldukça azdır. EA'ları farklılaştıran en önemli faktör kontrol parametrelerinin sayılarının azlığı, bir probleme uyarlanabilme dereceleri, kullandıkları yerel ve küresel arama operatörlerinin verimliliği ile harcadıkları hesaplama yükü ve kullandıkları yakınsama süresidir. Yaygın şekilde kullanılan pattern matrisi -veya sürü- tabanlı

EA'lar arasında PSO, ABC ve BSA sayılabilir.

Klasik optimizasyon algoritmalarının aksine EA'ların neden ve nasıl problemin alt en-iyilenmiş çözümlerinden birisine yakınsadığını açıklayan analitik bir kuram bulunmamaktadır. Bununla birlikte EA'ların neden sayısal bir problemin çözümüne yakınsadıklarını anlamak için kaos teorisinden analogik yaklaşımlarla çeşitli tahminlerde bulunma olanağı bulunmaktadır.

Kaos teorisinde bir problemin bir veya birden çok parametresinin sahip olduğu değerde meydana gelecek yeterince küçük bir değişim bile sonuçta veya ilerleyen hesaplama aşamalarında genliği belirsiz derecede bir değişime neden olabilir. Bu durum kelebek etkisi olarak adlandırılmaktadır. Kelebek etkisi, EA'larda pattern matrisinin başlangıçtaki yapısının, kontrol parametrelerinin başlangıç değerlerinin, iterasyon sayısının, kullanılan adım büyüklüğü üreticinin yapısının bir sonraki iterasyonda elde edilebilecek çözümlerin belirsizliği üzerindeki rolüne analogik olarak benzer.

Kaos teorisi bulunulan iterasyonda ilerleyen iterasyonlardan herhangi birisindeki ilgili problemin parametrelerinin alacağı değeri tahmin etmeye olanak vermez. Bu Kaos teorisinin tahmin-edilemezlik ilkesi olarak bilinir. Tahmin edilemezlik EA'ların problemin yeterince en-iyilenmiş bir alt çözümüne yakınsamayı güçleştiren en önemli nedendir. Kullanılan rassal çaprazlama/mutasyon operatörlerinin bir sonucu olan nümerik türbülansa dayalı yapısal stokastik doğaları EA'ların anlık davranışlarının ilerleyen aşamalarda nasıl evrileceğini -yerel bir çözüme takılıp takılmayacaklarını- tahmin etmeyi olanaksız kılar.

Kaos teorisi, yerel ölçekte -yüksek çözünürlükte- düzenli görülen çözümlerin, küresel ölçekte -düşük çözünürlükte- düzensizlik çözümler kümesinin bir elemanı gibi algılanabileceğini öngörür. Bu durum düzen/düzensizlik prensibi olarak bilinir. EA'larda yerel iterasyonlarda benimsenen yeni çözümlerin kabul süreci, aç-gözlü seçim gibi işleyişi iyi-tanımlanmış düzenli süreçlerdir. Oysa EA'larda yeni bir rassal çözüm üretme aşaması, EA'ların stokastik davranış patterleri nedeniyle düzensizdir. Düzen/düzensizlik EA'ların yerel çözüm arama operatörlerinin, küresel çözüm arama operatörlerine kıyasla daha yüksek çözünürlükte -sonuca görece daha yakın alanlarda arama yapmaları için- sanki daha düzenli arama yapıyormuş gibi görülmelerine de neden olur; çözümler daha az değişir. Oysa çözüm açısından değişen sadece çözünürlüktür. Yüksek çözünürlüklü rassal çözümler -yani kesin sonuca daha yakın çözümler- düşük çözünürlüklü rassal çözümlerden -görece kesin sonuca daha uzak çözümler- sanki daha düzenli gibi görünürler; bunun basit nedeni yerel çözümlerin standart sapmalarının görece düşük genlikli olmasıdır.

Kaos teorisinin katışıklık prensibi, çok-parametrelili problemlerde rassal çözümlerin ilgili problemin parametrelerinin katışık bir hali olabileceğini öngörür. EA'larda bir rassal çözüm ilerleyen hesaplama aşamalarında bulunduğu konumdan çok farklı bir konuma evrilebilir. Gerçekte -pattern matrisine çeşitliliği koruma amaçlı yeni rassal çözüm enjeksiyonu hariç- çoğu EA'da ilerleyen iterasyonlarda üretilen yeni rassal çözümler mevcut pattern matrisi elemanlarının farklı genliklerde rassal katışık hallerinden ibarettir. Dolayısıyla bu durum kaotik katışıklık prensibine analogik benzerlik gösterir.

Kaos teorisindeki en belirgin prensiplerden birisi geri-besleme kuralıdır. Bir çok kaotik fraktal pattern geri-beslemenin bir sonucudur. Belirgin şekilde BSA'da olduğu gibi kimi EA'larda bir sonraki iterasyona ait pattern matrisi bir önceki pattern matrislerinden beslenebilir. Veya bir sonraki pattern matrisleri önceki pattern matrisleri işlenerek üretilir; sonuç olarak çoğu EA'da patern matrisinin yeni jenerasyonlarını önermek için geri-besleme kullanılır.

İlgili problemin farklı çözünürlükteki rassal çözümleri, farklı fraktal desenler oluştururlar; yani bu çözümler açık şekilde ayrıştırılabilir. Bu durum kaos teorisinde fraktal davranış desenleri tanımlamaya benzer.

Yukarıda verilen kısa açıklamalar EA'ların stokastik doğalarının analogik olarak kaotik prensiplere önemli ölçüde uyumlu olduğunu göstermektedir. EA'lar stokastik, pattern matrisi -popülasyon; sürü - tabanlı, iteratif, rassal arama algoritmalarıdır. Temel işleyişleri büyük ölçüde kaotik prensipler kullanılarak açıklanabilir.

Bildirinin geri kalan bölümlerinde yeni bir EA olan BSA tanıtılmış ve onun farklı problem modellerine nasıl uyarlanabileceği üzerinde kısaca durulmuştur.

2.GERİ İZLEME EN-İYİLEME ALGORİTMASI (BSA)

BSA hafıza yapısına sahip, sürü tabanlı, bijektif formülü iteratif bir arama algoritmasıdır (Çivicioğlu, 2013a, 2013b, 2013c, 2013d). Standart BSA en-küçük değeri en-iyilemek için tasarlanmıştır. BSA'nın yapısı başlatma, doğal-seçim, değiştirme, çaprazlama ve güncelleme aşamalarından oluşur. Başlatma adımında P sürüsü rassal olarak $P = \{P_i = \langle p_{ij} \rangle | p_{ij} \sim U(a_j, b_j)\} | 1 \leq i \leq N, 1 \leq j \leq D$ tanımına uygun şekilde üretilir. Burada $U(a_j, b_j)$ a_j alt-sınırı ve

b_j üst-sınırı arasında tanımlanan eş dağılım fonksiyonunu göstermektedir. N ve D sırasıyla sürünün içerdiği rassal çözüm sayısını ve probleme ait parametrelerin sayısını göstermektedir. BSA'nın genel sistem denklemi t yineleme adımı için $P_{t+1} = P_t + \text{ölçek} \times M \square dx$ formuna uyar. Burada \square hadamart çarpımını göstermektedir. $M_{i,j} \in \{0,1\}$ 'dir. M_i 'leri üretmek için iki farklı strateji rassal sırayla kullanılır. İlk strateji rastgele konumlarda seçilen $D - \lceil q \times \kappa \times D \rceil$ $\kappa \sim U(0,1), 0 < q < 1$ adet parametrenin P_t ve P_{t+1} 'de farklı olmasını sağlar. İkinci strateji rassal olarak seçilen sadece bir parametrenin, P_t ve P_{t+1} 'de farklı olmasını sağlar. BSA'da *ölçek* değerini tanımlamak için problem yapısına uygun olarak herhangi bir istatistiksel dağılım kullanılabilir. dx arama doğrultusunu ifade eder. BSA'da doğal seçim süreci bir arama doğrultusu tanımlamak için kullanılacak olan sürüyü oluşturmayı amaçlar. Bu sürü P'nin erken evre yineleme adımlarından birisine ait sürünün saçılmış hali olabilir. Erken evre yineleme adımına ait sürünün ilk hali vektörel *permütasyon* kullanılarak üretilir. Herhangi bir yineleme adımında erken evre sürüsü rassal olarak mevcut sürüyle vektörel *permütasyon* kullanılarak güncellenir. Bu durum, BSA'ya kısa süreli bir hafıza yapısı kazandırmaktadır. BSA'nın yapısı Şekil 1'de yakıştırma-kod olarak verilmiştir (Çivicioğlu, 2013).

```

Input: ObjFun, N, D, maxcycle, mixrate, low1:D, up1:D
Output: globalminimum, globalminimizer
// rnd ~ U(0,1), rndn ~ N(0,1), w = rndint(·), rndint(·) ~ U(1,·) | w ∈ {1,2,3,...,·}
1 function bsa(ObjFun, N, D, maxcycle, low, up)
// INITIALIZATION
2 globalminimum = inf
3 for i from 1 to N do
4   for j from 1 to D do
5     Pi,j = rnd · (upj - lowj) + lowj // Initialization of population, P.
6     oldPi,j = rnd · (upj - lowj) + lowj // Initialization of oldP.
7   end
8   fitnessPi = ObjFun(Pi) // Initial-fitness values of P
9 end
10 for iteration from 1 to maxcycle do
// SELECTION-I
11 if (a < b) | a, b ~ U(0,1) then oldP := P end
12 oldP := permuting(oldP) // 'permuting' arbitrary changes in positions of two
individuals in oldP.
13 Generation of Trial-Population
// MUTATION
14 mutant = P + 3 · rndn · (oldP - P)
// CROSSOVER
15 map1:N,1:D = 1 // Initial-map is an N-by-D matrix of ones.
16 if (c < d) | c, d ~ U(0,1) then
17   for i from 1 to N do
18     | mapi,u(1:[mixrate·rnd·D]) = 0 | u = permuting({1,2,3,...,D})
19   end
20 else
21   for i from 1 to N do, mapi,randi(D) = 0, end
22 end
// Generation of Trial Population, T
23 T := mutant
24 for i from 1 to N do
25   for j from 1 to D do
26     | if mapi,j = 1 then Ti,j := Pi,j
27   end
28 end
// Boundary Control Mechanism
29 for i from 1 to N do
30   for j from 1 to D do
31     | if (Ti,j < lowj) or (Ti,j > upj) then
32     | | Ti,j = rnd · (upj - lowj) + lowj
33     | end
34   end
35 end
36 end
// SELECTION-II
37 fitnessT = ObjFnc(T)
38 for i from 1 to N do
39   if fitnessTi < fitnessPi then
40     | fitnessPi := fitnessTi
41     | Pi := Ti
42   end
43 end
44 fitnessPbest = min(fitnessP) | best ∈ {1,2,3,...,N}
45 if fitnessPbest < globalminimum then
46   globalminimum := fitnessPbest
47   globalminimizer := Pbest
// Export globalminimum and globalminimizer
48 end
49 end

```

Şekil 1. BSA'nın yakıştırma kodu.

BSA sayısal en-iyileme problemlerinin çözümünde oldukça etkili, görece çok kararlı -robust-, hızlı, iyi tanımlı, kısa ve anlaşılması çok kolay bir EA'dır.

3.PROBLEM MODELLEME

Bu bölümde çeşitli problemlerin EA kullanılarak nasıl çözülebileceği kısaca gösterilmiştir. EA olarak BSA kullanılmıştır. BSA veri girişi olarak amaç fonksiyonuna, varsa parametrelere ait alt-üst arama sınır değerlerine, iterasyon sayısına ve tek kontrol parametresi olan katıştırma oranı (mixrate) değerine ihtiyaç duyar.

Kümeleme problemi yaygın olarak kullanılan bir pattern tanıma yöntemidir. Yaygın olarak kullanılan kümeleme algoritmalarından biriside k-means kümeleme algoritmasıdır. k-means Denklem (1)'de tanımlanmıştır;

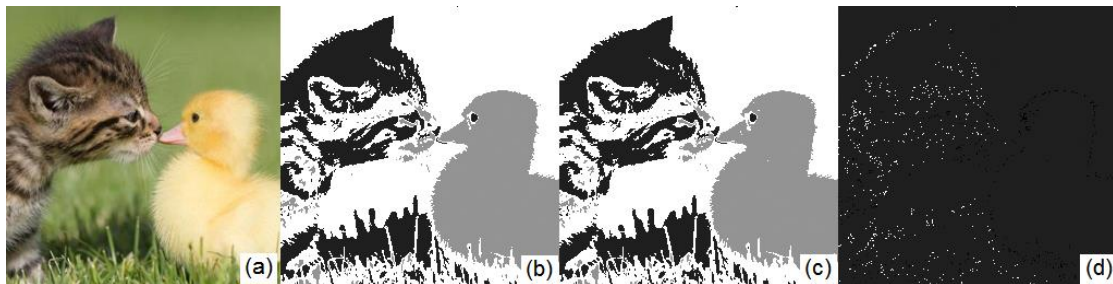
$$\operatorname{argmin}_c \sum_{j=1}^k \sum_{i=1}^n \|x_i^j - c_j\|^2 \quad (1)$$

Burada klasik yöntemlerle çözüm kolayca bulunabilir gibi gözükse de yüksek çözünürlüklü çok sayıda yerel çözüm olasıdır. Çoklu-çözüm nedeniyle evrimsel arama ile klasik çözümü oldukça geliştirmek mümkündür. Bu durumda önerilen klasik çözümün elde edilmesi ve Şekil 1'de tanımlanan $low_{1,D}$ ve $up_{1,D}$ limitlerinin klasik çözümü içerecek şekilde belirlenerek yakınsama için ihtiyaç duyulacak hesaplama süresinin azaltılmasıdır. Aynı durum Fuzzy C-Means ve SVM tabanlı kümeleme problemleri içinde geçerlidir. K-means için kullanılacak temel amaç fonksiyonu (Şekil 1'de $ObjFnc$) Denklem (1)'de verilen formdur.

Optimum küme sayısı Calinski-Harabasz gibi bir indeks kullanılarak tahmin edilebilirse de, küme merkezlerini çözmek her zaman kolay değildir. Calinski-Harabasz indeksi Test görüntüsü için hesaplandığında optimum küme sayısı 3 olarak elde edilmiştir. Test görüntüsü için k-means (30 tekrarlı çözümün en iyisi kullanılmıştır) ve BSA tarafından elde edilen küme merkezleri Tablo 1'de ve kümeleme sonuçları Şekil 2'de verilmiştir;

Tablo 1. K-means ve BSA ile küme sonuçlarının karşılaştırılması (s küme verisinin küme merkezine uzaklığı).

Küme Merkezi	K-means ($\sum s = 1.9391e+06$)			BSA ($\sum s = 1.9053e+06$)		
1	74.227	68.435	45.005	76.9317	68.974	48.839
2	235.676	212.376	154.123	240.501	218.159	160.763
3	136.611	143.694	89.372	135.867	148.842	88.238



Şekil 2 : Kümeleme sonuçları; (a) Test Görüntüsü, (b) K-Means, (c) BSA, (d) Oran Görüntüsü; (b)/(c).

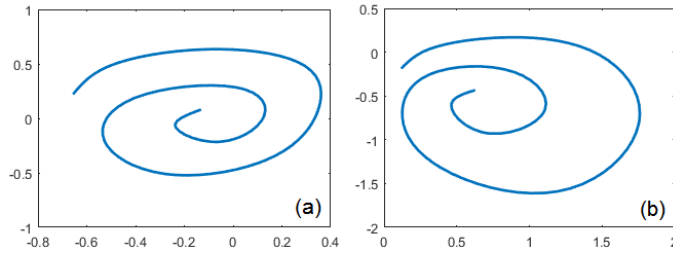
Tablo 1 incelendiğinde BSA'nın sonucu açık şekilde geliştirdiği görülmektedir.

Kamera kalibrasyonu bilgisayarlı görü uygulamalarında en sık karşılaşılan problemlerden birisidir. Denklem (2) kamera kalibrasyonu problemini tanımlamaktadır;

$$\operatorname{argmin}_{K,R,T} |K \times [R | T] \times X - U| \quad (2)$$

Gerçekleştirilen uygulamalar bu problem için herhangi bir $low_{1,D}$ ve $up_{1,D}$ tanımlamaya gerek olmadığını, problemin yeterli kontrol noktası koordinatı ve eşlenik görüntü koordinatları sağlandığında çözümün kolayca elde edilebildiğini göstermiştir. R 'nin ortogonalliğini sağlamak için parametrelerden R 'ye karşılık gelenler 3×3 matris halinde organize edilerek $[R, \sim, \sim] = SVD(R)$ dönüşümü yapılırsa R değerleri ayrıca işlenmeden ortogonal formda -amaç fonksiyonu için- elde edilebilir. İç yöneltme elemanlarını içerek K 'nin radyal ve mekez dışı distorsiyon değerlerini içerecek şekilde parametrelere ayrıca ayrıştırılması mümkündür. BSA grafikler üzerinde

affin veya projektif dönüklük ve öteleme değerlerini bulmak içinde kullanılabilir. Şekil 3 (a) ve Şekil 3 (b) arasındaki R dönüklük matrisinin EKK ve BSA ile elde edilen değerleri Tablo 2’de verilmiştir.

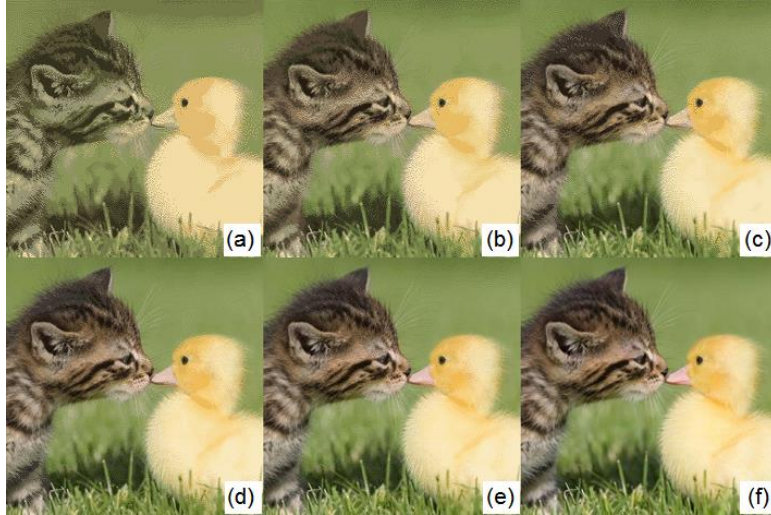


Şekil 3 . Aralarında R dönüklüğü olan iki grafik; (a) referans, (b) dönük eşlenik.

Tablo 2. Şekil 3’te verilen grafik için R tahmini.

R (Linear-EKK çözüm)			BSA		
1.555360	0.103053	0.927686	1.470585	0.252246	0.862978
-0.375185	1.409316	-0.681126	-0.333425	1.390716	-0.645602
0.000000	0.000000	1.000000	0.000000	0.000000	1.000000

Yarı-tonlama tabanlı bölgesel geçiş yumuşatma (image dithering), görüntü kuantalamada karşılaşılan yapay kenar bilgilerini yapay gürültü eklentisiyle azaltmayı sağlayan bir işlemdir. Dither uygulaması görüntü geliştirme, sıkıştırma, baskı üretimi, görüntü transferi, bant genişliği kazanma gibi bir çok farklı amaç için kullanılmaktadır. Şekil 4’de farklı seviyelerde kuantalanan bir görüntüye max-entropy tabanlı yarı-tonlama verilerinin bit-xor yöntemiyle gömülmesiyle elde edilen dither sonuçları verilmiştir. Bu durum kuantalanan görüntünün görsel algılanışını doğala yaklaştırmaktadır. Yarı-tonlama noktalarına ait görüntüdeki entropi’yi en-fazla yapan en-iyileme problemi BSA ile çözülmüştür.



Şekil 4. BSA tabanlı dither uygulaması ; (a) 8, (b) 16, (c) 32, (d) 64, (e) 128 ve (f) 256 seviyeli görüntüler.

Entropik dither fonksiyonunun hesaplanması aşağıda kısaca açıklanmıştır; s rassal çözümlerinin boyutu $k \times k \leftarrow \|s\|$ olmak üzere, $s := s \diamond (\text{ceil}(N/k) \times \text{ceil}(D/k))$, burada N ve D orijinal görüntü boyutları, k blok boyutudur. Kısaca blok boyutunda oluşturulan rassal çözüm s görüntü boyutlarına örneklenmektedir. Ardından orijinal görüntü I , s ’in min-max limitlerine ölçeklenir ve ölçeklenmiş imge O elde edilir. $Q = O > s$ ifadesinde $Q(Q = 1)$ koşulunu sağlayan piksellerin entropisi en-iyilenerek yarı-tonlama verisine ulaşılır.

Genelleştirilmiş 3D homojen koordinat dönüşümü Denklem (3) kullanılarak tanımlanmaktadır;

$$\arg \min_P \left| P_{4,4} \times X_{4,N} - X_{4,N}^* \right| \quad (3)$$

Burada $[R, \sim, \sim] = SVD(P_{1:3,1:3})$ dönüşümü rassal çözümlerden R ’ye karşılık gelenleri ortogonal bir R oluşturmaya zorlayacaktır. $T = P_{1:3,4}$ ’de tanımlanmıştır. $P_{4,4}$ genel ölçektir. Uygulama tecrübesi bu tür bir problemin

çözümünde BSA'nın başarısının doğal olarak X ve X^* 'in genel doğruluğundan etkilendiğini göstermiştir. Bu tip problemde sınır kullanımı performansı aşırı etkilememektedir.

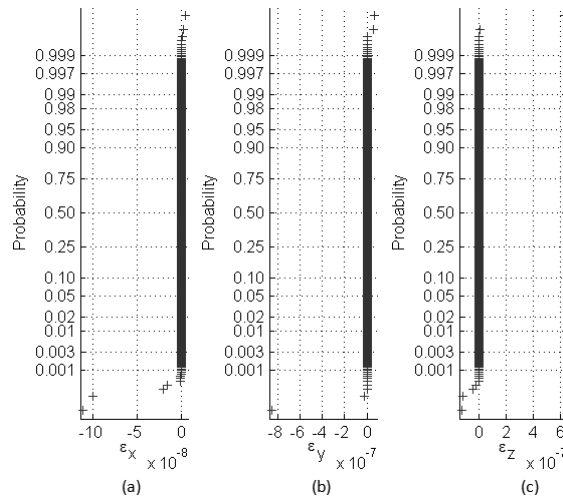
Jeosentrik-kartezyen koordinatları jeodezik koordinatlara dönüştürme problemi (Borkowski, 1987,1989; Bowring, 1976,1985, Chanfang ve Fei, 2010, Civicioğlu 2012) Denklemler (4,5) ile ifade edilen bir optimizasyon problemine dönüştürülebilir;

$$\varepsilon = \begin{bmatrix} (N+h) \cdot \cos \varphi \cdot \cos \lambda - x \\ (N+h) \cdot \cos \varphi \cdot \sin \lambda - y \\ N \cdot (1-e^2) + h \cdot \sin \varphi - z \end{bmatrix} \quad (4)$$

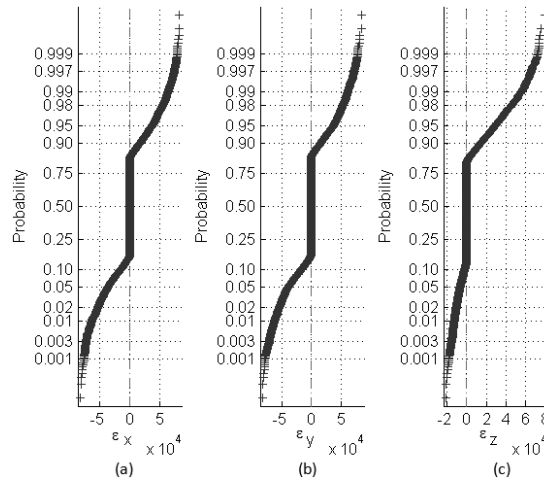
$$\arg \min_{0 \leq \lambda \leq \frac{\pi}{2}, h} \|\varepsilon\| \quad (5)$$

Bu problemin çözümünde kullanılan bir çok klasik yöntem bulunmakla birlikte, klasik çözümler çoğunlukla yer merkezi civarında yerel çözümlere takılma eğilimi göstermektedir. Bu nedenle BSA gibi bir EA kullanmak kararlı çözüm elde etmek için gereklidir. Uygulama tecrübesi bu problem için sınır tanımlamaları kullanımının BSA'nın çözüm bulma performansını aşırı etkilemediğini göstermiştir.

Jeosentrik-kartezyen koordinatları jeodezik koordinatlara dönüştürme problemi için gerçekleştirilen teste rassal olarak oluşturulan 3179 noktada Bowring metodu ve BSA kullanılarak gerçekleştirilen dönüşüm sonrası elde edilen hata değerlerinin normal dağılımları Şekil 5 ve Şekil 6'da verilmiştir.



Şekil 5. BSA kullanılarak gerçekleştirilen dönüşüm sonrası elde edilen $\varepsilon_{\{X,Y,Z\}}$ hatalarının normal-dağılımı; (a) X, (b) Y ve (c) Z.



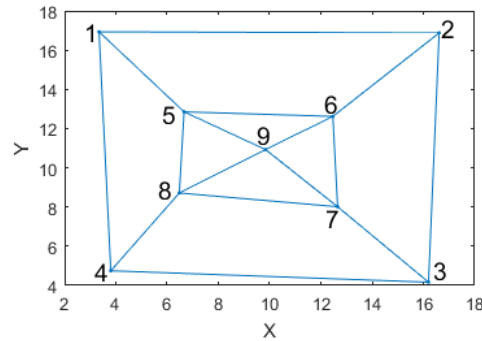
Şekil 6. Bowring metodu kullanılarak gerçekleştirilen dönüşüm sonrası elde edilen $\varepsilon_{\{X,Y,Z\}}$ hatalarının normal-dağılımı; (a) X, (b) Y ve (c) Z.

Elde edilen sonuçlar BSA'nın ilgili problemin çözümünde oldukça başarılı olduğunu göstermiştir.

Kenar-Ağ problemleri jeodezide karşılaşılan klasik bir problemdir. Bu problem Denklem (6)'da tanımlanmıştır;

$$\arg \min_{low \leq x \leq up} \sum |s_{ij} - \|x_i - x_j\||, \quad low = [x_0 \ y_0] - c \times [x_0 \ y_0], \quad up = [x_0 \ y_0] + c \times [x_0 \ y_0] \quad (6)$$

Burada s (i, j) arasındaki kenardır. $x = \langle X, Y \rangle$ 'dir. Bu problemin çözümünde yaklaşık koordinatları arama limitlerini tanımlamak için kullanılmak üzere hesaplama kalitesini belirgin şekilde arttırmaktadır. Şekil 7'de bir test kenar-ağ verilmiştir, bu ağa ait dayanak noktalarının 2D koordinatları ve BSA ile elde edilen hata değerleri Tablo 2'de görülmektedir. Uygulamada $c = 0.10$ kullanılmıştır.



Şekil 7. Test ağı.

Tablo 2. Test ağına ait kesin koordinatlar ve BSA tahmin hataları.

#	Test Ağı Koordinatları		Hata Değerleri	
	X	Y	ε_x	ε_y
1	3.295	16.870	0.000	0.000
2	16.566	16.778	0.000	0.000
3	16.119	4.068	0.000	0.000
4	3.724	4.682	0.000	0.000
5	6.602	12.780	0.000	0.000
6	12.408	12.532	0.000	0.000
7	12.581	7.925	0.000	0.000
8	6.407	8.641	0.000	0.000
9	9.777	10.848	0.000	0.000

Elde edilen sonuçlar BSA'nın kenar-ağı çok yüksek başarıyla dengeleyebildiğini göstermektedir.

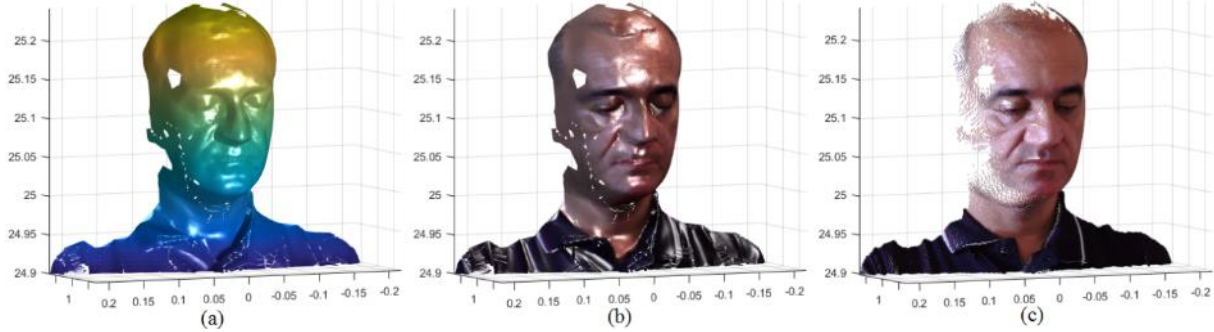
Nokta bulutlarının filtrelenmesinde iki temel yaklaşım bulunmaktadır; nokta-komşulukları tabanlı, mesh komşulukları tabanlı. Bir noktanın komşulukları kullanılarak hesaplanabilecek lineer yüzey Denklem (7) kullanılarak tanımlanır;

$$\arg \min_{a,b,d|c=1} \sum_{i=1}^n |a \cdot x_i + b \cdot y_i + cz_i - d| \quad (7)$$

Bu denklem kolayca çözülebilir gözüktüğü de EKK çözümü ve EA çözümler arasında farklar olabilmektedir. Bir $p=[u,v,w]$ noktasının bu yüzeydeki izi $-p_f = [x_f, y_f, z_f]$ - filtrelenmiş değer olarak kullanılabilir. B durumunda filtrelenmiş değerler Denklem (8) kullanılarak hesaplanabilir;

$$\begin{cases} x_f = u - a \cdot \frac{a \cdot u + b \cdot v + c \cdot w + d}{a^2 + b^2 + c^2} \\ y_f = v - b \cdot \frac{a \cdot u + b \cdot v + c \cdot w + d}{a^2 + b^2 + c^2} \\ z_f = w - c \cdot \frac{a \cdot u + b \cdot v + c \cdot w + d}{a^2 + b^2 + c^2} \end{cases} \quad (8)$$

Şekil 8’de BSA kullanılarak filtrelenmiş bir nokta bulutuna ait sonuçlar görselleştirilmiştir.



Şekil 8. BSA ile yüzey filtreleme; (a) Düzleme iz-düşürülmüş noktalardan oluşan yüzey, (b) (a) için yüzey normallerinin sürekliliği (görsel değerlendirme için doku verilmiştir), (c) kullanılan nokta bulutu.

4.SONUÇLAR

Bu bildiri EA’ların çözüm üretme nedenlerinin kaotik prensiplerle analogik benzerliği incelenmiş ve bir EA olan BSA’nın temel bazı geometik mühendisliği problemlerine nasıl uyarlanabileceği incelenmiştir. EA’lar sağladıkları esnek çözüm bulma avantajları nedeniyle farklı tipteki problemlerin sonuçlarını görece kolay kestirme yeteneğine sahiptir. Bu nedenle yaygın olarak kullanılmaktadırlar. Bu bildiri kısaca incelenen örnekler EA’ların geometik alanındaki problemlere kolaylıkla adapte edilebilecek çözüm arama algoritmaları olduklarını ayrıca vurgulamaktadır.

TEŞEKKÜR

Bu bildiri Tübitak tarafından 115Y235 kodlu araştırma projesi kapsamında desteklenmiştir.

KAYNAKLAR

- Borkowski, K.M.**, 1987, Transformation Of Geocentric To Geodetic Coordinates Without Approximations, *Astrophysics and Space Science* 139, 1–4.
- Borkowski, K.M.**, 1989, Accurate Algorithms to Transform Geocentric to Geodetic Coordinates, *Bulletin Ge’ode’sique*, 63, 50–56.
- Bowring, B.R.**, 1976, Transformation From Spatial To Geographical Coordinates, *Survey Review*, 23, 323–327.
- Bowring, B.R.**, 1985, The Accuracy Of Geodetic Latitude And Height Equations, *Survey Review*, 28, 202–206.
- Chanfang Shu , Fei Li**, 2010, An Iterative Algorithm To Compute Geodetic Coordinates, *Computers & Geosciences*, 36 1145–1149.
- Civicioglu, P.**, 2012, Transforming Geocentric Cartesian Coordinates to Geodetic Coordinates by Using Differential Search Algorithm, *Computers and Geosciences*, 46, 229-247.
- Civicioglu, P.**, (a) 2013, Circular Antenna Array Design by Using Evolutionary Search Algorithms, *Progress In Electromagnetics Research B*, 54, 265-284.
- Civicioglu, P.**, (b) 2013, Backtracking Search Optimization Algorithm for numerical optimization problems, *Applied Mathematics and Computation*, 219, 8121–8144.

Civicioglu, P., (c) 2013, Artificial cooperative search algorithm for numerical optimization problems, *Information Sciences*, 229, 58–76.

Civicioglu, P., Besdok, E., (d) 2013, A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, *Artificial Intelligence Review*, 39 (4), 315-346.