

Suyun Hareketinden Esinlenilerek Geliştirilen Hesapsal Zekâ Yöntemlerinin İncelenmesi

Feyza Altunbey Özbay¹, Bilal Alataş²

^{1,2}Fırat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ.

e-posta:faltunbey@firat.edu.tr, balatas@firat.edu.tr

Özet

Literatüre kazandırılmış çok başarılı arama ve optimizasyon algoritmaları olsa da; bilimsel alanda sürekli iyileşme ve daima daha iyiyi arama felsefesi altında yeni tekniklerin tasarlanması, geliştirilmesi ve uygulanması önemli bir görevdir. Ayrıca her türlü arama ve optimizasyon problemleri için en iyi sonucu veren algoritma henüz tasarlanmadığından dolayı; sürekli, yeni hesapsal zekâ optimizasyon algoritmaları önerilmekte ya da var olanların daha etkili çalışması için öneriler sunulmaktadır. Özellikle son yıllarda bu bilinçle araştırmacılar, yeni hesapsal zekâ optimizasyon algoritmalarını literatüre kazandırmışlar ve başarılı uygulamalar gerçekleştirmişlerdir. Önerilen algoritmaların çoğunun biyolojik ve fizik tabanlı olduğu söylenebilir. Bu çalışmada da suların hareketinden esinlenilerek geliştirilmiş üç algoritma olan Su Akış Algoritması, Su Çevrim Algoritması ve Su Akış Benzeri Algoritma; ilk kez detaylı olarak birlikte incelenmiş ve bu algoritmaların iyi sonuç verdiği optimizasyon problemlerinin tipleri hakkında analiz yapılmıştır. Literatürde fazla bilinmeyen ve fazla uygulanmamış bu algoritmaların farklı varyantları ve optimize edilmiş parametreleriyle bir çok arama ve optimizasyon problemlerinde etkili sonuçlar vereceği beklenmektedir.

Anahtar kelimeler

Hesapsal zekâ yöntemleri; Su hareketi; Optimizasyon.

Review of Computational Intelligence Method Inspired from Behavior of Water

Abstract

Though the very successful search and optimization algorithms have been gained to literature; the design, development, and implementation of new techniques are an important task under philosophy of continuous improvement and always searching better in scientific fields. Furthermore, the single algorithm which provides the best results for every search and optimization problems hasn't yet designed. That is why, continuously new computational intelligence optimization algorithms have been proposed or suggestions or additions to existing ones have been presented for effective solutions. Especially in recent years, with this awareness, researchers have gained new computational intelligence optimization algorithms to the literature and they have performed successful implementations. The most of the recommended algorithms are biological and physics-based. In this study, three algorithms; Water Flow Algorithm, Water Cycle Algorithm, and Flow Like Water Algorithm, which are improved inspiring by the movements of waters have been examined in detail for the first time and analysis about the types of optimization problems which these algorithms give promising results is made on. In the literature, with different variants and optimized parameters of these relatively unknown and unimplemented algorithms, effective results have been expected to provide in search and optimization problems.

Keywords

Computational intelligence methods; Water movement; Optimization.

© Afyon Kocatepe Üniversitesi

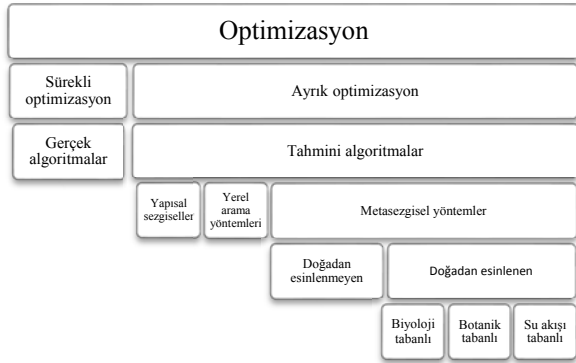
1. Giriş

Metasezgisel algoritmalar her türlü probleme uygulanabilmektedir. 'Meta' kelimesi üst düzey veya ötesinde anlamına gelirken, 'Sezgisel' kelimesi, deneme yanılma yoluyla keşfetmek ve aramak anlamına gelmektedir. Metasezgisel

algoritmalar bir amaç fonksiyonu için, aday çözümler geliştirerek iteratif bir şekilde optimal çözümleri arar. Bu algoritmalar iki temel işlemden oluşmaktadır. Bunlar çözüm arama ve çözüm işleme şeklindedir. Bu iki süreç, uygun hesaplama zamanında optimum veya yakın optimal çözümler

aramak için iteratif olarak uygulanır. Çözüm arama bulunan çözümlerin çeşitliliğini arttırmakta ve yerel optimum çözümlerin bulunmasına yardımcı olmaktadır. İşleme süreci, elde edilen çözümlerin yakınsamasını sağlayarak sonuçların kalitesini arttırmaktadır. Metasezgisel algoritmaların performansı bu iki işlem arasındaki uygun kombinasyona bağlıdır (Hieu, 2011).

Metasezgisel optimizasyon yöntemleri birkaç on yıldır genellikle doğrusal olmayan matematik veya tam ve reel sayı karışım program modelleri içeren kompleks gerçek dünya problemlerinin çözümünde kullanılmaktadır. Metasezgisel algoritmalar, doğadan ilham alan ve doğadan ilham almayan algoritmalar olmak üzere ikiye ayrılmaktadır. Doğa yüzyıllardır gelişmektedir ve bu özelliği ile metasezgisel algoritmaların tasarımında yaratıcı fikirler vermektedir. Optimizasyon için kullanılan algoritmalar ve alt başlıkları Şekil 1'de şematik olarak gösterilmiştir (Hieu, 2011).



Şekil 1. Optimizasyon algoritmalarının sınıflandırılması.

Doğadan esinlenen algoritmalar biyolojiden, botanikten ve su akışından ilham alan algoritmalar olmak üzere üç alt başlık altında toplanabilmektedir. Biyolojiden ilham alan farklı algoritmalar bulunmaktadır. Bu algoritmalar evrimsel algoritmalar ve sürü algoritmaları olarak sınıflandırılabilir. Evrimsel algoritmalar genetik algoritma, memetik algoritma, yapay bağışıklık algoritmaları gibi algoritmalarlardır. Sürü tabanlı algoritmaları ise parçacık sürü, ateşböceği, karınca koloni, balık sürüsü ve yarasa algoritmalarıdır (Akyol and Alatas2012).

Botanikten esinlenerek oluşturulan algoritmaları,

istilacı ot ve aşılardan esinlenen algoritmalar oluşturmaktadır (Can and Alatas 2015). Su akışı ile ilgili doğanın süreçlerinden etkilenen bir takım algoritmalar bulunmaktadır. Bu teknikler su akış modellerine dayalı metin görüntülerini işlemede kullanılmaktadır. Bunlar nehir sistemlerinin dinamiklerine dayanarak oluşturulan Akıllı Su Damlası Algoritması (Shah-Hosseini, 2009), sıvı akışlarının davranışına dayanan Su Akışı Benzeri Algoritma (Yang and Wang 2007), Su Akış Algoritması (Hieu and Ng2011), Su Çevrim Algoritması (Sadollah et al. 2015), Su Buharlaştırma Optimizasyonu (Kavehand Bakhshpoori2016), Benzetimli Yağmur Damlası Algoritması (Ibrahim et al. 2014), Nehir formasyon Dinamiği Algoritması (Rabanal et al, 2010), Su Dalga Optimizasyonu (Zeng, 2015), Çevrimsel Su Dalga Algoritması (Colak and Varol 2015), Yapay Dış Algoritması (Ali et al, 2015)'dir. Her ne kadar biyoloji tabanlı çok algoritma olsa da; sadece su davranışından esinlenerek geliştirilen algoritmaların da fazla sayıda olduğu görülmektedir. Bu çalışma ile ilk kez bu algoritmaların üçü bir arada araştırmacılara tanıtılmış ve deneysel sonuçlara bağlı olarak iyi sonuçların alındığı optimizasyon problemlerinin tipleri hakkında analiz yapılmıştır.

2. Su Akış Algoritması

Su Akış Algoritması (SAA) meteorolojideki hidrolojik çevrim, doğadaki suyun akış özellikleri ve doğadaki erozyon olgusundan esinlenerek geliştirilmiş metasezgisel bir yöntemdir. Suyun akış özelliğinin her zaman düşük şiddetten başlayıp giderek yükselmesi, yağmur damlalarının topraktaki birçok alana yayılması, suyun akışının yerde bıraktığı erozyon gücü gibi süreç, olay ve konular temel alınır (Hieu, 2011, Hieu and Ng 2011).

SAA'da kullanılan terimler ve aralarındaki ilişkiler için dünyadaki hidrolojik çevrimlerin aşamaları bilinmelidir. Bunlar kısaca aşağıda listelenmiştir:

- Buharlaştırma: Okyanustaki suyun ısı etkisiyle atmosfere gaz olarak aktarılması.
- Yoğunlaşma: Atmosferdeki su buharının gaz halindeyken sıvı ya da katı hale dönüşmesi.
- Taşınma: Okyanustaki su buharının rüzgâr

etkisiyle karaya hareket etmesi.

- Yağış: Atmosferdeki su buharının yoğunlaşması ile ortaya çıkan ve dünyanın yüzeyine kadar inebilen her türlü ürün.
- Terleme: Bitkiler ve bitki örtüsü tarafından, yer yüzeyinden atmosfere aktarılan su buharı.
- Yeraltı Suyu: Yeryüzüne inen yağıştan sonra, toprakta kalan suyun toprağın altına geçerek tekrar okyanusa ulaşması.

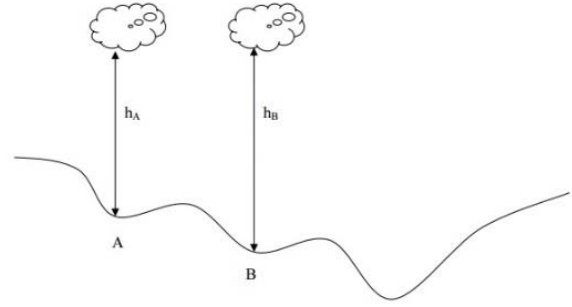
SAA'da, su damlasının global optimum noktasını bulması için kullanılan üç tane bellek listesi vardır. Bunlar:

- En İyi Konum Listesi (PO-list): Bulunan en iyi noktaları saklar.
- Aşınmamış Konum Listesi (UE-list): Erozyon ihtimali olmayan, aşınmamış konumların yerel optimum noktalarını saklar.
- Aşınmış Konum Listesi (E-list): Erozyon süreci başladıktan sonra erozyonun araziye ne ölçüde yumuşatacağına bakarak optimum noktaları saklar.

SAA meteorolojideki hidrolojik çevrimlerden ve su akış erozyonundan esinlenerek gerçekleştirilmiştir. Tek amaçlı optimizasyon problemlerini çözerken hidrolojik çevrimin keşif aşamasında sadece taşınma ve yağış bilgileri kullanılmaktadır. Ancak çok amaçlı optimizasyon problemlerini çözerken; keşif aşamasında buharlaşma, yoğunlaşma, terleme ve yeraltı suyu bilgileri kullanılmaktadır. Burada kullanılan bilgilerin sayısı ne olursa olsun problemin çözüm yöntemi çok fazla değişikliğe uğramaz (Hieu, 2011, Hieu and Ng 2011).

Su akış erozyonunun keşif aşamasını simülasyon oluşturmaktadır. Optimum noktalara ulaşmak veya bulunan noktadan daha iyi olan bir optimum noktaya ilerlemek su akış erozyonunun keşif aşamasını oluşturur. Şekil 2'de görüldüğü üzere B noktasında erozyon olma ihtimali A noktasında erozyon olma ihtimalinden daha fazladır.

Hidrolojik çevrim ve su akış erozyonunun arasındaki ilişki bize erozyon süreci için maksimum yineleme sayısını vermektedir. Erozyon süreci için maksimum yineleme sayısı Denklem 1'deki gibi ifade edilir.



Şekil 2. Rakımın erozyon gücüne etkisi.

$$MaxUIE = \varphi_1 Q(\pi^*) + \varphi_2^{LBz} (1)$$

$MaxUIE$: Erozyon süreci için maksimum yineleme sayısı.

φ_1 : Yağışın etkisini ve düşme kuvvetini temsil eden bir parametre.

φ_2 : Yağışın etkisini ve düşme kuvvetini temsil eden bir parametre.

$Q(\pi^*)$: Yerel optimum pozisyondaki suyun damla sayısı.

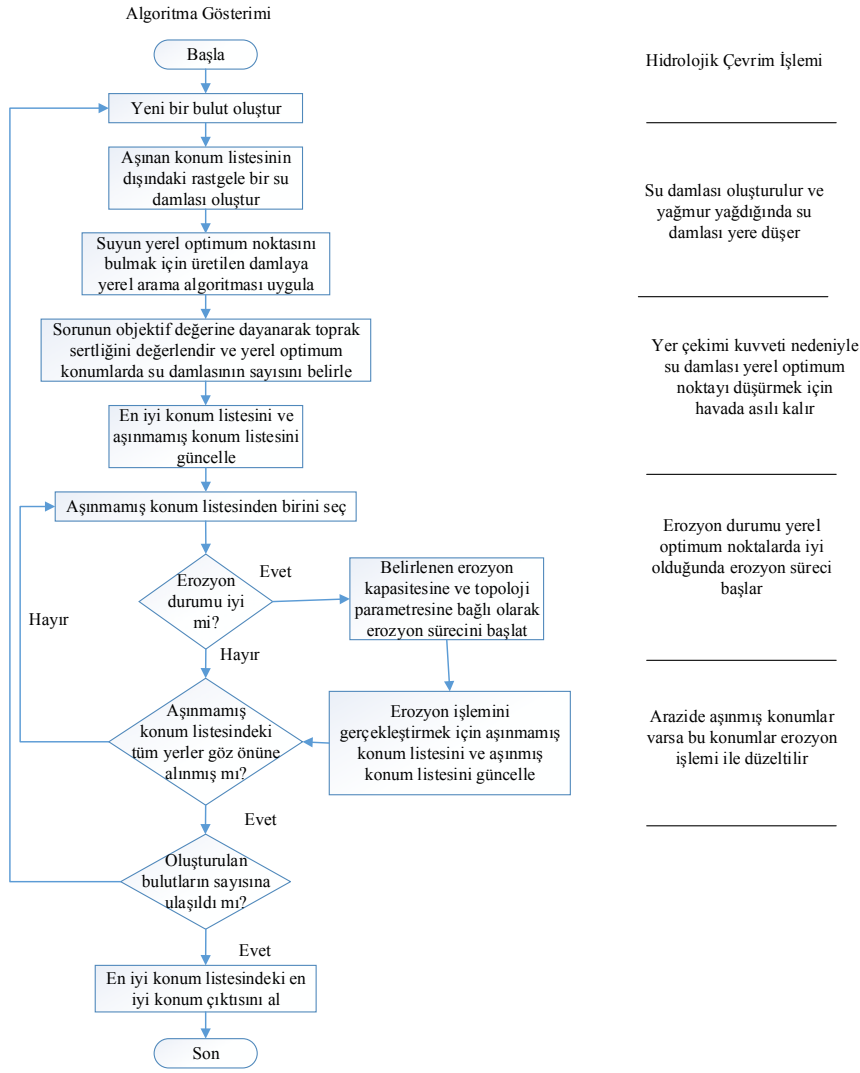
LB : Alt sınır.

z : Aşınmış konum listesindeki objektif bir değer.

Büyük vadilerdeki merkez noktalar küçük vadilerdeki merkez noktalara göre daha kolay bulunmaktadır. Bu, büyük vadilerin merkez noktalarındaki su damlası sayısının küçük vadilerinkine oranla daha fazla olacağı anlamına gelir. Bu mantıkla algoritmadaki erozyon süreci modellenmiştir.

SAA'nın akış şeması Şekil 3'te gösterilmiştir (Hieu, 2011, Hieu and Ng 2011).

Bu algoritma daha çok görüntü işleme alanında uygulanmıştır. Kim ve arkadaşları optimizasyon sorunları çözmek için su akış özelliklerini kullanan ilk yazarlardır (Kim et al. 2012).



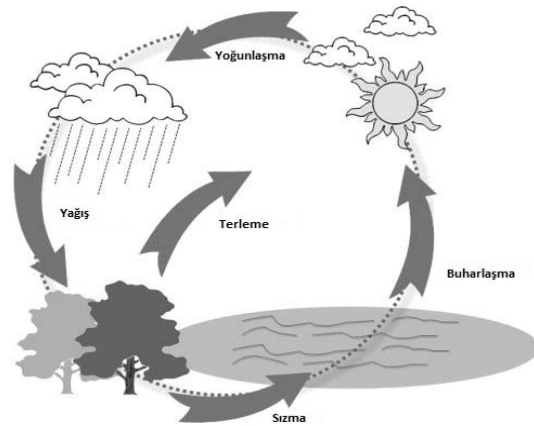
Şekil 3.SAA'nın akış seması.

3. Su Çevrim Algoritması

Su Çevrim Algoritması (SÇA); kısıtlı optimizasyon ve mühendislik tasarım problemlerinin çözümü için önerilen optimizasyon tekniğidir. SÇA'nın ana fikri doğadan esinlenilmiştir ve gerçek dünyada dere ve nehirlerin yokuş aşağı denize nasıl aktıklarının ve su devrinin gözlenmesi üzerine kurulmuştur.

Su bir yerden başka bir yere yokuş aşağı aktığı zaman, bir nehir ya da bir dere oluşturur. Yani çoğu nehirler kar ya da eski buzulların eridiği dağlarda oluşmaktadır. Nehirler her zaman aşağı doğru akar. Onların aşağı iniş ve denizde biten yolculukları sayesinde sular yağmur ve derelerden toplanır. Şekil 4 su devrinin bir bölümü için basitleştirilmiş bir şemadır. Bitkiler fotosentez sırasında su verirken, nehir ve göllerdeki su buharlaşır. Buharlaşan su soğuk atmosferde yoğunlaşacak

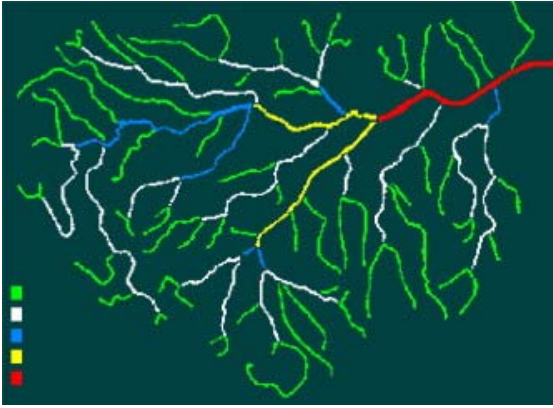
bulutları oluşturmak için atmosfere taşınır; bu su yağmur ve yağış şeklinde dünyaya geri bırakılır. Bu süreç su çevrimi olarak adlandırılır (Sadollah et al. 2015, Eskandar et al. 2012).



Şekil 4. Su çevriminin basitleştirilmiş şeması.

Gerçek dünyada, karlar eridikçe ve yağmur

yağdıka, suların çođu yer altı su tabakasına (akifer) girer. Yer altı su rezervlerinin birçok alanı vardır. Yer altı su tabakası yer altı suyu olarak da adlandırılır (Şekil 4'te sızma oku ile gösterilmiştir). Daha sonra akiferdeki su kırsal bölge altına akar. Yer altı suyu nehre ya da bataklıđa tahliye edilebilir. Bitki ve diđer yeşilliklerden buhar halinde nem verilmesine ek olarak; su dere ve nehirlerden buharlaşır. Sonuç olarak; bu döngü devam ettikçe bulutlar oluşur ve yağmurlar meydana gelir. Şekil 5 derelerin nehre, nehirlerin denize nasıl aktığını şematik olarak göstermektedir. Bu şekil ağacı ve ağacın köklerini andırmaktadır (Sadollah et al. 2015, Eskandar et al. 2012).



Şekil 5. Akıntılarının nasıl olduğunu gösteren sistematik şema (Eskandar et al. 2012).

Yeşil renk ile gösterilen akıntılar birinci dereceden akıntılardır. Bunların birleşmesi beyaz renk ile gösterilen ikinci dereceden akıntılar oluşturur. İkinci dereceden akıntılarının birleşmesi ile de mavi renk ile gösterilen üçüncü dereceden akıntılar oluşmaktadır. Böylelikle nehirlere kadar ve nihayetinde denize kadar akmaya devam eder.

Diđer metasezgisel algoritmalara benzer şekilde, önerilen metot “yağmur damlaları” diye adlandırılan bir ilk popülasyon ile başlar. İlk olarak, yağmur ya da yağışa sahip olduğumuz farz edilir. En iyi birey (en iyi yağmur damlası) deniz olarak seçilir. İyi yağmur damlalarının bir kısmı nehir olarak seçilir ve kalan kısmı nehre ve denize akacak dere olarak düşünülür.

Akışın büyüklüğüne bağlı olarak, her bir nehir derelerden su emer. Aslında, nehre ve/veya denize giren bir deredeki suyun miktarı diđer derelerden farklıdır. Buna ilaveten; nehirler en çok yokuş aşağı

konumlu denize akar.

Popülasyon tabanlı metasezgisel metotlar kullanarak bir optimizasyon problemini çözmek için, soru deđişkenlerinin deđerleri bir dizi olarak oluşturulması gereklidir. Bu dizi SÇA'da bir tek çözüm için “yağmur damlası” olarak adlandırılır. Bir N boyutlu optimizasyon probleminde, bir yağmur damlası $1 \times N_{değ}$ 'lık bir dizidir. Bu dizi aşağıda gösterildiđi şekilde tanımlanır.

$$Yağmur\ Damlası = [x_1, x_2, x_3, \dots, x_n](2)$$

Optimizasyon algoritmasına başlamak için, $N_{pop} \times N_{değ}$ boyutlarında yağmur damlalarının bir matrisini gösteren bir aday üretilir. Sonuç olarak; rastgele oluşturulmuş X matrisi ortaya çıkar (sıra: popülasyon sayısı; sütun: tasarım deđişkenleri sayısı).

Yağmur Damlası Popülasyonu =

$$\begin{bmatrix} Yağmur\ Damlası_1 \\ Yağmur\ Damlası_2 \\ \vdots \\ Yağmur\ Damlası_{N_{pop}} \end{bmatrix} = \begin{bmatrix} x_1^1 & \dots & x_{N_{değ}}^1 \\ \vdots & \ddots & \vdots \\ x_1^{N_{pop}} & \dots & x_{N_{değ}}^{N_{pop}} \end{bmatrix} (3)$$

Karar deđişkeni deđerlerinin $(x_1, x_2, x_3, \dots, x_{N_{var}})$ herbiri sürekli ve ayrık problemler için sırası ile gerçel sayılar ya da ön tanımlı bir küme olarak tanımlanabilir. Bir yağmur damlasının maliyeti aşağıda verilen maliyet fonksiyonunun (M) hesaplanması ile elde edilir:

$$M_i = M_i = f(x_1^i, x_2^i, \dots, x_{N_{değ}}^i) \quad i = 1, 2, 3, \dots, N_{pop} \quad (4)$$

Denklemdaki N_{pop} =ilk popülasyonda yağmur damlası sayısını, $N_{değ}$ =Tasarım deđişkenlerinin sayısını belirtir. İlk adım olarak N_{pop} tane yağmur damlası oluşturulur. En iyi bireylerden (minimum deđerler) bir takımı (N_{dn}) deniz ve nehirler olarak seçilir. Bu seçilen bireyler arasındaki en küçük deđere sahip yağmur damlası deniz olarak kabul edilir. Popülasyonda geriye kalanlar ise ya nehirlere akacak ya da direk olarak denize akacak dereleri oluşturur.

$$N_{dn} = Nehirlerin\ sayısı + 1(Deniz)(5)$$

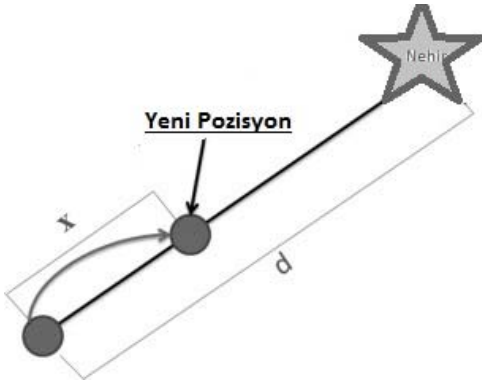
$$N_{\text{yağmur damlaları}} = N_{\text{pop}} - N_{\text{dn}}(6)$$

Yağmur damlasının akış yoğunluğuna bağlı olarak nehirlere ve denize devretmek için, Denklem 7 kullanılır.

$$ND_n = \text{yuvarla} \left\{ \left\lfloor \frac{\text{Maliyet}_n}{\sum_{i=1}^{N_{sr}} \text{Maliyet}_i} \right\rfloor \times N_{\text{yağmur damlaları}} \right\}, n = 1, 2, 3, \dots, N_{\text{dn}}(7)$$

Denklemde verilen ND_n belirli nehirlere ve denize akacak derelerin numarasıdır.

Dereler yağmur damllarından oluşmaktadır ve birbirleri ile birleşerek nehirleri oluşturmaktadır. Aynı zamanda derelerin bazıları direkt olarak denize akmaktadırlar. Bütün nehirler ve dereler denizde sonlanmaktadır (en iyi optimal nokta). Şekil 6 derelerin belirlenmiş nehre akışının sistematik görüntüsüdür (Eskandar et al. 2012).



Şekil 6. Derelerin belirlenmiş nehre akışının görüntüsü.

Şekil 6'da gösterildiği gibi, bir dere Denklem 8 ile verilen rastgele uzaklığı kullanarak nehir ile arasındaki bağlantı hattı boyunca nehre doğru akar.

$$X \in (0, C \times d), C > 1(8)$$

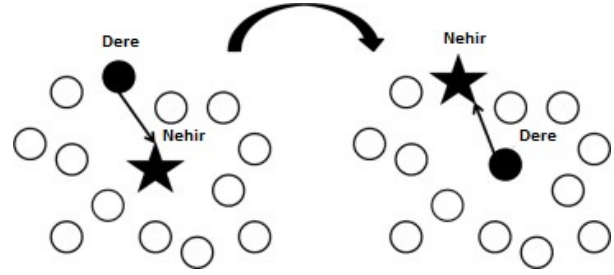
C , 1 ve 2 arasında bir değerdir. C için en iyi değer 2 olarak seçilebilir. Dere ve nehir arasındaki mevcut mesafe d olarak temsil edilir. Denklemdeki X 'in değeri 0 ile $C \times d$ arasında bir dağıtık rastgele sayıya (homojen ya da herhangi bir uygun dağılım olabilir) karşılık gelir. Birden büyük olan C 'nin değeri derelerin nehirlere doğru farklı yönlerde akmalarını sağlar. Aynı zamanda bu kavram nehirlerin denize akışında da kullanılabilir.

Sonuç olarak dereler ve nehirler için yeni pozisyon şöyle verilebilir:

$$X_{Dere}^{i+1} = X_{Dere}^i + rand \times C \times (X_{Nehir}^i - X_{Dere}^i)(9)$$

$$X_{Nehir}^{i+1} = X_{Nehir}^i + rand \times C \times (X_{Deniz}^i - X_{Nehir}^i)(10)$$

$rand$ sayısı 0 ile 1 arasında homojen dağıtılmış rastgele bir sayıdır. Bir dere tarafından verilen çözüm onun bağlandığı nehirden daha iyi ise eğer, nehir ve derenin pozisyonları değiştirilir (dere nehir olur, nehir dere olur). Böyle bir değişim benzer şekilde nehirler ve deniz için de yapılabilir. Şekil 7 diğer dereler ve aktığı nehir arasında en iyi çözüm olan derenin değişimini göstermektedir (Sadollah et al. 2015, Eskandar et al. 2012).



Şekil 7. En iyi olan derenin nehir ile değişimi.

Buharlaştırma algoritmayı erken yakınsamadan koruyabilen en önemli faktörlerden biridir. Doğada görülebileceği gibi, bitkiler fotosentez esnasında suyu doğaya bırakırken su nehir ve göllerden buharlaşır. Buharlaşan su soğuk atmosferde yoğunlaşacak bulutları oluşturmak için atmosfere taşınır; bu su yağmur ve yağış şeklinde dünyaya geri bırakılır. Bu yağışlar yeni dereler oluşturur ve yeni dereler denize akacak yeni nehirlere akar. Bu döngü su devri olarak adlandırılır. Önerilen metotta, dere ve nehirler denize aktıkça buharlaştırma işlemi deniz suyunun buharlaşmasına neden olur. Bu varsayım yerel optimum değerlerde sıkışıp kalmayı önlemek için önerilmiştir. Aşağıdaki sözde kod nehrin denize akıp akmayacağını nasıl tespit edileceğini göstermektedir.

$$\text{if } |X_{Deniz}^i - X_{Nehir}^i| < d_{maks} \quad i = 1, 2, 3, \dots, N_{dn} \\ - 1 \text{ Buharlaştırma ve yağış işlemi}$$

S₁: Son

d_{maks} 0'a yakın küçük bir sayıdır. Nehir ve deniz

arasındaki mesafe d_{maks} daha küçük ise, bu nehrin denize varacağını gösterir. Bu durumda buharlaşma işlemi uygulanır ve doğada görüldüğü gibi yeterli buharlaşma sonrası yağmur (yağış) başlar. d_{maks} içinküçük bir değer denize yakın arama yoğunluğunu teşvik ederken, büyük bir değer ise denize yakın aramayı azaltır. Bundan dolayı d_{maks} denize yakın arama yoğunluğunu kontrol eder (optimal çözüm). d_{maks} 'ın değeri ayarlamalı olarak azalır:

$$d_{maks}^{i+1} = d_{maks}^i - \frac{d_{maks}^i}{\max \text{ yineleme}} \quad (11)$$

Buharlaşma işlemi yerine getirildikten sonra, yağış işlemi uygulanır. Yağış işleminde, yeni yağmur damlaları farklı yerlerde dereleri oluşturur (Genetik algoritmadaki mutasyon operatörüne benzer bir işlemdir). Yeni oluşturulan derelerin yeni lokasyonlarını oluşturmak için, Denklem 12 kullanılır.

$$X_{Dere}^{yeni} = LB + rand \times (UB - LB) \quad (12)$$

LB ve UB verilen problemde tanımlanan alt ve üst sınırlardır.

Yeniden, oluşan yağmur damlalarının en iyisi denize akan nehir olarak kabul edilir. Kalanların ise nehre ve direkt olarak denize akacak dereleri oluşturduğu farz edilir.

Kısıtlı problemlerde hesaplama performansını ve yakınsama oranını artırmak için, sadece direkt olarak denize akan dereler için Denklem 13 kullanılır. Kısıtlı problemler için uygun bölgelerde denize yakın aramayı (optimum sonuç) geliştirmek için, bu denklem denize direkt olarak akan derelerin jenerasyonlarını teşvik etmeyi amaçlar.

$$X_{Dere}^{yeni} = X_{Deniz} + \sqrt{\mu} \times random(1, N_{değ}) \quad (13)$$

μ denize yakın arama bölgesinin aralığını gösteren katsayıdır. Random normal dağılım gösteren rastgele bir sayıdır. μ için değer yükseldikçe aramanın uygun bölgeden çıkma olasılığı artar. Diğer yandan, μ küçük seçilmesi algoritmayı denize yakın daha küçük bölgede aramaya sevk eder. μ için

uygun değer 0.1 olarak ayarlanır. Matematiksel bakış açısında, μ terimi standart sapmayı gösterir ve bu yüzden μ varyans kavramını tanımlar. Bu kavramları kullanarak μ varyans ile üretilen bireyler belirlenen en iyi nokta(deniz) etrafında dağılır.

SÇA'nın ele aldığı arama uzayında, dereler ve nehirler hem problemin belirli sınırlamalarını hem de tasarım değişkenlerinin sınırlarını ihlal edebilir. Bu metotta, modifiye edilmiş uygulanabilir tabanlı dört kurala dayalı bir mekanizma kısıtlamaları idare etmek amacıyla kullanılmıştır.

Kural-1: Herhangi uygun bir çözüm herhangi bir imkânsız bir çözüme tercih edilir.

Kural-2: Kısıtlamaların hafif ihlalini içeren mümkün olmayan çözümler (ilk tekrarda 0.01'den son tekrar 0.001'e kadar) uygun çözüm olarak görülür.

Kural-3: İki uygun çözüm arasında, daha iyi amaç fonksiyonuna sahip olan tercih edilir.

Kural-4: İki uygun çözüm arasında, kısıtlama ihlalinin daha küçük toplamına sahip olan çözüm tercih edilir.

Birinci ve dördüncü kuralları kullanarak, arama mümkün olmayan bölgelerden ziyade mümkün bölgelere odaklanır. Üçüncü kuralı uygulamak iyi cevaplarla birlikte uygun bölgelerde araştırmaya imkân sağlar. Çoğu yapısal optimizasyon problemleri için global minimum uygun tasarım sınırının uzayına yakın ya da o uzayın üzerinde yer alır. İkinci kuralı uygulayarak dereler ve nehirler sınırlara yaklaşır ve yüksek muhtemelle global minimuma ulaşılabilir.

Sonlandırma kriteri için, genellikle diğer metasezgisel algoritmalarda olduğu gibi, maksimum iterasyon sayısı, işlemci zamanı ya da son iki sonuç arasında izin verilen, küçük negatif olmayan bir tolerans değeri kullanılabilir. Yakınsama kriterini sağlamak için SÇA maksimum iterasyon sayısına kadar çalıştırılabilir (Sadollah et al. 2015, Eskandar et al. 2012).

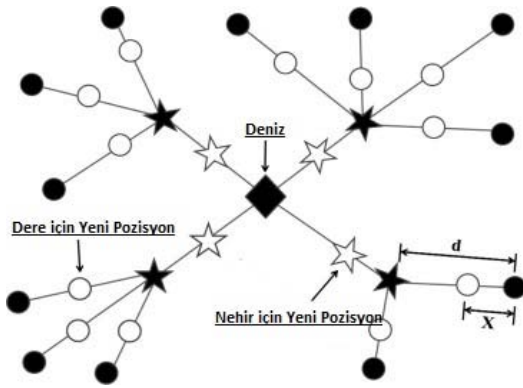
3.1. SÇA'nın adımları

SÇA'nın adımları Şekil 8'de özetlenmiştir.

Adım-1: SÇA'nın ilk parametrelerini belirle: $N_{sr}, d_{max}, N_{pop}, maks_iterasyon$
Adım-2: İlk popülasyonu üret ve (3), (5), ve (6).denklemleri kullanarak deniz, nehirler ve dereleri oluştur
Adım-3: (4).denklemleri kullanarak her bir yağmur damlasının maliyet değerini hesapla
Adım-4: (7).denklemleri kullanarak nehirler ve deniz için akışın yoğunluğunu belirle
Adım-5: (9). denklemleri kullanarak dereleri nehirlere aktır
Adım-6: (10).denklemleri kullanarak nehirleri en yokuş aşağı olan denize aktır
Adım-7: Şekil 7'de gösterildiği gibi en iyi sonucu veren dere ile nehrin pozisyonlarını değiştir
Adım-8: Adım-7'ye benzer şekilde nehir denizden daha iyi bir sonuç bulursa, deniz ile nehrin pozisyonlarını değiştir
Adım-9: Buharlaştırma sözde kodunu kullanarak buharlaştırma şartını kontrol et
Adım-10: Buharlaştırma şartı sağlanırsa, (12). ve (13).denklemleri kullanarak yağış işlemini gerçekleştir

Şekil 8. SÇA'nın adımları.

SÇA'nın sistematik görüntüsü Şekil 9'da gösterilmiştir. Çemberler, yıldızlar ve elmas sırasıyla dereler, nehirler ve denize denk gelmektedir. Boş şekiller ise dereler ve nehirlerin yeni pozisyonlarını göstermektedir (Sadollah et al. 2015, Eskandar et al. 2012).



Şekil 9. SÇA'nın sistematik görüntüsü.

4. Su Akışı Benzeri Algoritma

Su Akışı Benzeri Algoritma (SABA) da suyun akışından ilham alınarak önerilmiş bir arama ve optimizasyon algoritmasıdır. Çözüm suyun akışına benzer olarak modellenir ve çözüm uzayındaki amaç fonksiyonu akışın araziden geçişi gibi modellenir. SABA sıvı akış davranışından ilham alınarak geliştirilmiştir. Sıvı, hareketini yerçekimi kuvveti ve enerji korunum kanunlarına bağlı olarak yapar. Çözüm aramaları başladığında sadece bir su

akışı yayılımı vardır. Sonra, arazi bölündüğünde veya engebeli bir alanla karşılaşıldığında akış alt akımlara bölünür (Wu et al. 2010). Bunun aksine akışlar aynı yerde toplandıkları zamanda tek bir akış olarak birleşir. Böylece akışların sayısı optimal çözüm için araştırıldığında değişir.

Su akışları aday çözüm olarak görüldüğünde ve amaç fonksiyonunun yüzeyinde arazi planlandığında, daha aşağı pozisyondaki suyun hareketi en iyi çözüm araması için dikkate alınır. Su akışlarının bazı doğal davranışları SABA'nın tasarımında kabul görür (Yang and Wang 2007; Wu et al. 2010). SABA tasarlanırken aşağıdaki aşamalar takip edilir.

- Yerçekimi ve enerji korunum kanunlarına bağlı olarak sıvı aşağı doğru akış davranışı gösterir. Böylece çözüm aramaları alt iterasyondan üst iterasyona doğru sürekli tekrarlanır.

- Akışkan hareketi ile su engebeli araziye ters yönde taşınır. Bir akışkan engebeli arazi ile karşılaştığı zaman alt akımlara bölünür ve onun hareketi bölünen parçaları aşar. SABA bir aday çözümün ayrılmasını bir akışkanın iki alt akıma bölünmesine benzer şekilde modeller. Hızlı bir akışkan daha yavaş bir akışkana göre daha fazla alt akım oluşturacaktır. Bir akış sınırlı bir hızla sahip değilse alt akım sayısı hesaplanamaz. Bu sebeple akış hızı sınırlı bir akışkan kullanılmalıdır.

- Kinetik ve potansiyel enerjinin korunum kanununa göre akışkanın en alttaki davranışı ile en üstteki davranışı farklıdır. Buna bağlı olarak akışkan en altta daha fazla alt akımlara ayrılır. En alt pozisyonda hata yapmaktan kaçınılması gerekir, çünkü SABA su akışının kötü lokasyonlara ve geniş bilinmeyen bölümlere gitmesine izin verebilir.

- Tüm akışlar aynı yere çıktıklarında, hepsi birleşerek tek bir akım olur. Çoklu akımlar aynı yere taşındığında SABA'nın aday çözüm sayısı azalır, bu yüzden gereksiz aramalardan kaçınmak gerekir.

- Su akışları, atmosferdeki suyun buharlaşma miktarına da bağlıdır. Burada buharlaşan su yağmurla tekrar yeryüzüne iner. SABA'da akan suyun bir bölümü manuel olarak buharlaşan su miktarı olarak düşünülür. Yağış işlemi ise doğal

yağışlara bağlı olarak uyarlanır.

İlk olarak rasgele oluşturulan yerde bir su akışı belirlenir. Başlangıç kütlesi ve hızı da girilmelidir. Başlangıç akışının yeri optimizasyon problemi için ilk çözümdür. Akışkanın hızı ve potansiyel enerjisi dikkate alınarak yeni bir çözüm üretilir.

SABA'da akışkanın taşınması değişken grup problemlerinin çözümünde akışkanın taşınacağı en iyi komşu lokasyona sürekli adımlara taşınacak şekilde belirlenir. Akıştaki çeşitli taşıma stratejileri farklı optimizasyon problemleri için kullanılabilir (Yang and Wang 2007, Wu et al. 2010). Akışkanın hızı, akışkanın yeni gideceği lokasyonu belirlemek için kullanılmaz fakat taşınıp taşınmadığını belirlemek için kullanılır. Akışın alt akımlara ayrılması hareket yeteneğinin bir sonucudur. Yüksek hareketli bir akışkan düşük hareketli bir akışkana göre daha fazla alt akımlara ayrılır. Ayrılmış alt akımın lokasyonu orijinal çıkışın komşu lokasyonundan yola çıkarak türetilir. Aranılan son lokasyonun yasak bölgesi, mümkün olmayandan kurtulmak için ve ayarlanan komşu lokasyondan geçen bölge kontrol edilir. Mümkün olan komşu lokasyonlar değerlendirilir ve sıralanır. Bir akışkan eğer n tane alt akıma ayrılırsa, n tane sıralanmış akışkandan ilki buradaki alt akım olarak atanır. Bir akışkan hiçbir alt akıma ayrılmazsa, burada akan akışkan mümkün olan en iyi koşu lokasyona akar. Bunun aksine eğer bir akışkanın hızı sıfır ise o lokasyonda durur. Farklı bir akışkanla birleşene kadar hareketsiz olarak kalır ya da buharlaşır ve daha sonra yeryüzüne geri döner (Yang and Wang 2007).

Enerji korunum kanununda; SABA akışı alt akımlara ayrıldığında her alt akım için kütle ve hız ataması yapılmalıdır. Orijinal akışkanın kütlesi bölünür ve alt akımlara ayrılır. Her alt akımın hızı orijinal akıştan alt akımlara doğru artma veya azalmaya bağlı olarak hesaplanır. Her iterasyonda orijinal akıştan alt akımlara doğru ayrıldığı zaman bu alt akımlar farklı akımlar gibi davranır.

Genellikle, sezgisel algoritmaların çoğu, yeni çözüm uzayları keşfetmek veya istenmeyen lokasyonlardan kurtulmak için silme veya değiştirme uygular. Doğal davranışların taklidi için

ve geniş çözüm uzaylarının keşfi için SABA'ya buharlaşma ve yoğunlaşma eklenir. Su akışındaki su atmosferde buharlaşır. Buharlaşan su sürekli birikir, düşük akış hızlı yeni bazı akışlara katılır. Aday çözümler SABA'da su akışı gibi modellenir ve alt akım sayısına göre dinamik değişiklik gösterir. SABA'nın dört temel değişkeni vardır, bunlar; akışın ayrılması ve taşınması, akışın birleşmesi, suyun buharlaşması ve yoğunlaşmasıdır (Yang and Wang 2007, Wu et al. 2010).

Akışın Ayrılması ve Taşınması: Kinetik enerjisi yüksek olan bir akışın daha düşük kinetik enerjiye sahip olan bir akıştan daha fazla alt akımlar oluşturur. Yerçekimi kuvveti ve enerji korunum kanuna bağlı olarak suyun akışı yeni lokasyonlara doğrudur. Doğada çoklu akımlar için su çağlayanları yukarıdan aşağıya doğru akar. SABA'da bir akış aday çözüm olarak görülür ve başarılı çözüme ulaşmak için bu akış alt akımlara ayrılır. Akışkan hareketi çözüm aramak için mevcut lokasyondan yeni lokasyona doğru akar. SABA'da akışkanın taşınma işleminin tasarımı probleme dayalıdır. Farklı problem tiplerine göre farklı akış şemaları hazırlanır.

Akışın Birleşmesi: İki ya da daha fazla akış aynı lokasyona hareket ettiğinde tek bir akımda birleşirler. Lokasyon etrafındaki çözümü güçlendirmek için birleşen akımların kütle hareketleri toplanır. İstenmeyen lokasyondan kurtulmak için birleşen akış hareketli enerjiden durgun hale getirilebilir. Bu işlemle bir akışın başka bir akışla aynı lokasyonu paylaşıp paylaşmadığı sistematik olarak kontrol edilir. Eğer birleşiyorsa sonraki akış farklı bir akışla birleşir. Nadiren, birleşme işleminden sonra akışkanlar sıkışabilir ve istenmeyen durumlar ortaya çıkabilir, bu durumda akış durur.

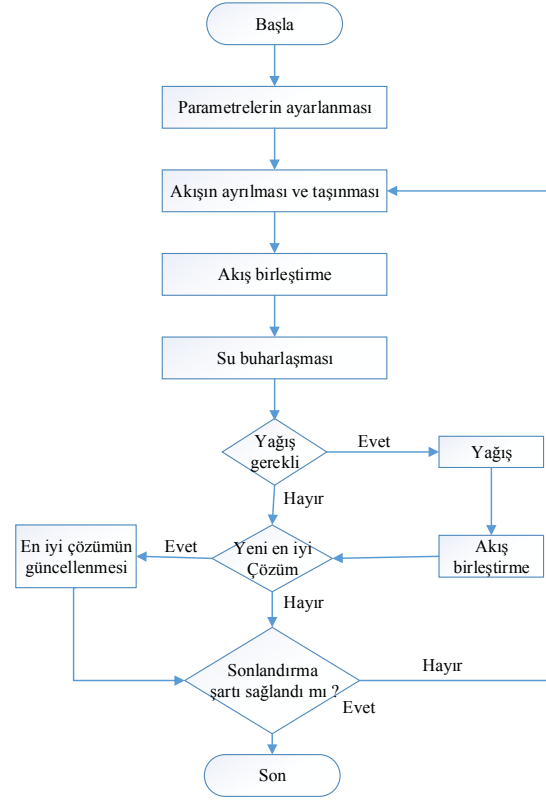
Suyun Buharlaşması: Suyun buharlaşması her bir su akışına bağlıdır. Akıştaki suyun bir kısmı buharlaşarak atmosfere gider. Buharlaşan suyun miktarı tekrardan yoğunlaşana kadar kümülatif olarak tutulur. Bu işlemle, çözüm aramalarını genişletmek için akışın yenilenmesi sağlanır. Suyun buharlaşması işleminde tüm akışın kütlesi güncellenir

Yoğunlaşma: Bu işlem doğadaki yağışlara benzer olarak periyodik olarak gerçekleşir. Yeni akış popülasyonu yağışları oluşturur ve aranan çözüm uzayında yeni alanların keşfedilmesi sağlanır. Nadiren, tüm akışlar durduğunda mevcut akış yoğunlaşmayla hareketlenir. Kabul edilebilir bir çözüm elde edilene kadar, her bir hesaplanan iterasyon şartlı olarak gerçekleşir ve buradaki dört işlem yinelemeli olarak yapılır. Suyun doğal yaşam döngüsüne benzemek için SABA’da iki çeşit yoğunlaşma gerçekleşir. İlki, durgun akış için hareket enerjisi kullanan zorunlu yoğunlaşma, ikincisi düzenli yoğunlaşmadır. Zorunlu yoğunlaşma tüm akışların hızı sıfır olduğunda uygulanır. Atmosferdeki buharlaşmanın tamamlanması için akışların tamamı zorlanır daha sonra mevcut akışların sayısı değişmeksizin yeryüzüne geri döner. Buna rağmen dönüş lokasyonları orijinal lokasyondan stokastik sapma gösterir. Düzenli yoğunlaşma, buharlaşarak yeryüzünden uzaklaşan suyun tekrar geri dönebilmesi için periyodik olarak uygulanır. Yeni akışlar içindeki bu işlemde yeni çözüm kümeleri bulunur.

Zorunlu ve düzenli yoğunlaşma işlemlerinin her ikisinde de dikkat edilmelidir. Aynı lokasyona çoklu akışlar atanabilir. Bu yüzden, gereksiz akışları elimine etmek için yoğunlaşma işleminden sonra akış birleştirme işlemi gerçekleşir (Yang and Wang 2007).

SABA için gerekli parametrelerin içeriği: iterasyon sınırı, P ; başlangıç kütlesi, M_0 ; hız, V_0 ; orijinal akıştaki temel hareket, T ; alt akım sayısının sınırı, n ; üst sınır, $b_h^{(u)}$; alt sınır, $b_h^{(l)}$; komşu adım uzaklığı, r_h ; yağış dengesi, e_h ; her bir koordinat, h ; iterasyon sayısı, t ’den oluşur. SABA’nın varsayılan sonlandırma şartı, iterasyon sayısının p ’ye ulaşmasıdır. Alt akımların sayısı, bir akışkanın temel hareketinin bölünmesi ile belirlenir. Başlangıç hızı ve kütlesi kullanıcı tarafından atanır: $M_0, V_0 = 2T \sim 3T$ arasında önerilir. Alt akım sayısı bir akışkanın üst sınırını belirler. $n = 3$ önerilir. SABA’nın akış hareket işlemi hedeflenen probleme bağlıdır. Komşu lokasyonların yerleri için sürekli adım boyutu belirlenir. Daha küçük adım daha hassas sonuçlar elde etmemizi sağlar. Genelde $r_h = 1$ belirlendiğinde

tüm olası komşu çözümleri kapsar. titerasyon için periyodik yağış işlemi gerçekleşir, akış yeniliğinin taşmasını engellemek için, $t > P/20$ önerilir. Önerilen yağış $e_h = b_h^{(u)} - b_h^{(l)}$, tüm çözüm uzayını kapsar. Şekil 10, SABA’nın akış diyagramını göstermektedir (Yang and Wang 2007).



Şekil 10.SABA’nın akış diyagramı.

5. Sonuçlar

Tüm problemler için en iyi sonucu veren algoritmanın bulunamaması ve araştırmacıların daima en iyiyi bulma istekleri dolayısıyla sürekli yeni metasezgisel algoritmalar önerilmektedir. Önerilen algoritmaların çoğunun biyolojik ve fizik tabanlı olduğu söylenebilir. Ancak pek incelenmeyen ve suların hareketinden esinlenilerek geliştirilmiş etkili algoritmalar da bulunmaktadır. Bu makalede çok güncel olan ve ileride popüler hale gelmeye başlaması çok muhtemel su davranışını temel alan yapay zekâ optimizasyon algoritmaları ilk kez birleştirilip tanıtılmıştır. Özellikle bunlardan Su Akış Algoritması, Su Çevrim Algoritması ve Su Akış Benzeri Algoritma; ilk kez bu çalışmada detaylı olarak birlikte incelenmiştir.

Literatürde fazla bilinmeyen ve fazla uygulanmamış bu algoritmaların farklı varyantları ve optimize edilmiş parametreleriyle bir çok arama ve optimizasyon problemlerinde etkili sonuçlar vereceği beklenmektedir. Farklı arama ve optimizasyon problemleri için ve özellikle çok amaçlı, farklı tipte karar değişkeni içeren, matematiksel modeli çıkarılamayan ve çıkarılsa bile çözümü çok uzun zaman alan kompleks problemlere etkili sonuçlar verebilecek alternatif yöntemler olarak görülmektedir. Ayrıca, algoritmaların etkinliğinin doğaları gereği dağıtık ve paralel versiyonları ile arttırılabileceği öngörülmektedir.

Kaynaklar

- Hieu,T.T., 2011. A water flow algorithm for optimization problems. Doctorate Thesis, National University of Singapore, 234.
- Akyol,S.,Alataş, B.,2012. Güncel sürü zekâsı optimizasyon algoritmaları. *Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, **1**, 36-50.
- Can, Ü.,Alataş, B.,2015. Bitki zekâsında yeni bir alan: kök kütlesi optimizasyonu. *Türk Doğa ve Fen Dergisi*, **4**, 8-14.
- Shah-Hosseini,H., 2009. The intelligent water drops algorithm: A Nature-Inspired Swarm-Based Optimization Algorithm. *International Journal of Bio-Inspired Computation*, **1**, 71–79.
- Yang, F.C.,Wang, Y.P.,2007. Water flow-like algorithm for object grouping problems. *Journal of the Chinese Institute of Industrial Engineers*,**24**, 475–488.
- Hieu,T.T.,Ng, K.M.,2011. A water flow algorithm for flexible flow shop scheduling with intermediate buffers. *Journal of Scheduling*, **14**, 483-500.
- Sadollah, A.,Eskandar, H.,Bahreininejad, A.,Kim, J.H. 2015. Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems, *Applied Soft Computing*, **30**, 58–71.
- Kaveh, A. and Bakhshpoori, T., 2016. A new metaheuristic for continuous structural optimization: water evaporation optimization. *Structural and Multidisciplinary Optimization*. 1-21.
- Ibrahim, A., Rahnamayan, S., and Martin, M.,2014. Simulated raindrop algorithm for global optimization. *Proceedings of 27th IEEE Canadian Conference on Electrical and Computer Engineering, Canada*, 1-8.
- Rabanal,P., Rodriguez, I.,Rubio, F.,2010. Applying river formation dynamics to the steiner tree problem. *Proceedings of 9th IEEE International Conference on Cognitive Informatics (ICCI), Beijing*, 704-711.
- Zheng, Y.J.,2015. Water wave optimization: a new nature-inspired metaheuristic. *Computers & Operations Research*, **55**, 1-11.
- Colak,M.E., Varol, A., 2015. A novel intelligent optimization algorithm inspired from circular water waves, *Elektronika ir Elektrotehnika*, **21**, 3-6.
- Ali, J.,Saeed, M.,Chaudhry,N.A.,Luqman, M.,Tabassum, M.F.,2015. Artificial showering algorithm: a new meta-heuristic for unconstrained optimization. *Science International*, vol. **27**, 4939-4942.
- Kim,I.K., Jung., D.W.,Park, R.H.,2012. Document image binarization based on topographic analysis using a water flow model. *Pattern Recognition*, **35**, 265-277.
- Eskandar, H.,Sadollah, A., Bahreininejad, A., Hamdi, M., 2012. Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems, *Computers & Structures*, **110**, 151-166.
- Wu,T.H., Chung, S.H.,Chang, C.C.,2010. A water flow-like algorithm for manufacturing cell formation problems, *European Journal of Operational Research*, **205**, 346–360.