



Kanalizasyon optimizasyonu için kısıt yönetimi mekanizmalarının karşılaştırılması

Comparison of constraint-handling mechanisms for sewer optimization

Mustafa Erkan Turan¹ , Tülin Çetin^{2,*} , Mümin Emre Şenol³ 

^{1,2} Manisa Celal Bayar Üniversitesi Mühendislik Fakültesi, İnşaat Mühendisliği Bölümü, Manisa, Türkiye

³ Manisa Celal Bayar Üniversitesi Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, Manisa, Türkiye

Öz

Bu çalışmanın amacı kanalizasyon hidrolik tasarım optimizasyonu probleminde en uygun kısıt yönetimi mekanizmasını araştırmaktır. Bu amaçla problemin çözümünde daha önce kullanılmamış olan Karahindiba Optimizasyon Algoritması kullanılmıştır. Ele alınan problem kısıtlı optimizasyon problemi olduğundan, problemin çözümünde kısıt yönetimi mekanizmalarına ihtiyaç duyulmaktadır. Çalışmada çeşitli kısıt yönetimi mekanizmalarının, algoritmanın performansı üzerindeki etkileri iki farklı boyuttaki kanalizasyon şebekesinde araştırılmıştır. Kullanılan kısıt yönetimi mekanizmaları statik ceza yöntemi, dinamik ceza yöntemi, olurlu çözümlerin üstünlüğü ile birleştirilmiş dinamik ceza yöntemi, üstel dinamik ceza yöntemi, olurlu çözümlerin üstünlüğü ile birleştirilmiş üstel dinamik ceza yöntemi, eklektik ceza yöntemi, uyarlanabilir ceza yöntemi ve ters tanjant yöntemidir. Performans ölçütü olarak çözüm kalitesi, çözüm süresi ve gelişme oranı ölçütleri kullanılmıştır. Ele alınan kısıt yönetimi mekanizmalarından en tutarlı sonuca sahip olan yöntem, uyarlanabilir ceza yöntemi olmuştur.

Anahtar kelimeler: Kanalizasyon optimizasyonu, Karahindiba optimizasyon yöntemi, Kısıtlı optimizasyon, Kısıt yönetimi mekanizmaları, Ceza yöntemi

1 Giriş

Bir kanalizasyon şebekesinin ana bileşenleri borular ve muayene bacalarıdır. Bu bileşenlerin sayısı, konum ve boyut değerleri şebekenin inşa edileceği yerleşim bölgesinin büyüklüğüne, imar planına, topoğrafyasına, nüfusuna ve tasarım kriterlerine göre projelendirme aşamasında belirlenmektedir. Kanalizasyon şebekelerinin işlevini yerine getirmesinin yanında en uygun maliyete de sahip olması amaçlanmaktadır. Böylece kanalizasyon optimizasyonu kavramı ortaya çıkmaktadır.

Kanalizasyon optimizasyonunda en uygun maliyeti elde edebilmek için proje mühendisi çoğunlukla kendi deneyimlerinden faydalanır ve deneme yanılma yöntemini kullanır. İzlenen bu yol, sonucun en uygun maliyete sahip olduğunu garanti etmemektedir. Dolayısıyla elde edilen sonuç kişiye bağlı olmaktadır. Günümüzde bu dezavantaj, en uygun maliyete sahip şebekeyi elde edebilmek için

Abstract

The aim of this study is to investigate the most appropriate constraint-handling mechanism in the sewer hydraulic design optimization problem. For this purpose, Dandelion Optimizer, which has not been used before to solve this problem, was used. Since the considered problem is a constrained optimization problem, constraint-handling mechanisms are required to solve this problem. In this study, the effects of various constraint-handling mechanisms on the performance of the algorithm were investigated in two different sized sewer networks. The constraint-handling mechanisms used are static penalty method, dynamic penalty method, dynamic penalty method with the superiority of feasible points, exponential dynamic penalty method, exponential dynamic penalty method with the superiority of feasible points, eclectic penalty method, adaptive penalty method and inverse tangent method. Solution quality, solution time and progress ratio were used as performance criteria. The most consistent result among the constraint-handling mechanisms discussed, was the adaptive penalty method.

Keywords: Sewer optimization, Dandelion optimizer, Constrained optimization, Constraint-Handling mechanisms, Penalty method

optimizasyon yöntemleri kullanılarak giderilmeye çalışılmaktadır.

Kanalizasyon sistemi tasarımı, şebeke planının belirlenmesi ve şebekenin hidrolik tasarımı olmak üzere iki farklı kısımdan oluşmaktadır. Şebeke planının belirlenmesi, muayene bacalarının, boruların, pompa istasyonlarının, arıtma tesisi ve diğer tesislerin konumunun dolayısıyla akış yönlerinin belirlenmesini içerir. Hidrolik tasarım ise boru çapları ile boru eğimlerinin belirlenmesidir.

Literatür incelendiğinde, kanalizasyon optimizasyonu hakkındaki çalışmalarda şebeke planının optimizasyonunun ve hidrolik tasarım optimizasyonunun ayrı ayrı veya birlikte ele alındığı görülmüştür. Ancak çoğunlukla hidrolik tasarım optimizasyonu hakkında çalışmalar [1-9] bulunmaktadır. Şebeke planının optimizasyonu hakkındaki çalışmalar [10-14] ise daha azdır. Bazı çalışmalarda ise şebeke planı

* Sorumlu yazar / Corresponding author, e-posta / e-mail: tulin.cetin@cbu.edu.tr (T. Çetin)

Geliş / Received: 23.06.2023 Kabul / Accepted: 09.11.2023 Yayınlanma / Published: 15.01.2024

doi: 10.28948/ngumuh.1318989

optimizasyonu ve hidrolik tasarım optimizasyonu [15-18] birlikte yapılmaktadır.

Optimizasyon problemleri kısıt durumuna göre kısıtlı ve kısıtsız olarak iki gruba ayrılmaktadır. Kısıtsız optimizasyon problemlerinde karar değişkenlerinin alabileceği değerlerin bir sınırlaması yoktur. Öte yandan kısıtlı optimizasyon problemlerinde karar değişkenlerinin alabileceği değerler sınırlandırılmıştır. Optimizasyon problemlerinin çözümünde, problemin yapısına göre farklı çözüm yöntemleri kullanılmaktadır. Metasezgisel optimizasyon yöntemleri, bu çözüm yöntemlerinden biridir. Metasezgisel optimizasyon yöntemleri genellikle kısıtsız optimizasyon problemlerinin çözümüne yönelik geliştirilmektedirler. Bu nedenle metasezgisel optimizasyon yöntemlerine kısıtlı optimizasyon problemlerini çözme yeteneği kazandırmak için, farklı kısıt yönetimi mekanizmaları kullanılmaktadır.

Tan vd. [19] kanalizasyon optimizasyonu probleminin çözümünde statik ceza yöntemini, Çunkaş ve Ürkmez [20] dalgıç asenkron motorların tasarım optimizasyonu probleminin çözümünde eklektik ceza yöntemini, Garip vd. [21] kısıtlı sürekli optimizasyon problemleri için dinamik ceza yöntemini, Baykasoğlu ve Özsoydan [22] mekanik dizayn optimizasyon problemleri için ters tanjant yöntemini, Taşgetiren [23] oryantiring problemi için uyarlamalı ceza yöntemini, İyengar ve Sigman [24] kayıp ağların kontrolü için üstel dinamik ceza fonksiyonu yöntemini, Gölcük [25] mühendislik tasarım problemlerinin çözümü için olurlu çözümlerin üstünlüğü ile birleştirilmiş dinamik ceza yöntemini ve olurlu çözümlerin üstünlüğü ile birleştirilmiş üstel dinamik ceza yöntemini kullanmıştır.

Literatürde farklı problemler için kısıt yönetimi mekanizmalarının çözüme etkisinin araştırıldığı çalışmalar [25-27] bulunmakla beraber, kanalizasyon hidrolik tasarım optimizasyonu problemi konusunda bir çalışmaya rastlanmamıştır.

Bu çalışmada kanalizasyon hidrolik tasarım optimizasyonu problemi, literatüre yeni kazandırılmış ve bu problemin çözümünde daha önce kullanılmamış bir metasezgisel optimizasyon algoritması olan Karahindiba Optimizasyon Algoritmasıyla (KOA) çözülmüştür. Kanalizasyon optimizasyon probleminin kısıtlarını ele almak için kısıt yönetimi mekanizmaları kullanılmıştır. Kullanılan kısıt yönetimi mekanizmaları sırasıyla, statik ceza yöntemi, dinamik ceza yöntemi, olurlu çözümlerin üstünlüğü ile birleştirilmiş dinamik ceza yöntemi, üstel dinamik ceza yöntemi, olurlu çözümlerin üstünlüğü ile birleştirilmiş üstel dinamik ceza yöntemi, eklektik ceza yöntemi, uyarlanabilir ceza yöntemi ve ters tanjant yöntemidir.

Yapılan çalışmanın amacı, farklı kısıt yönetimi mekanizmalarının kanalizasyon hidrolik tasarım optimizasyonu probleminin çözüm performansına etkisini araştırmaktır. Bu sebeple KOA ile iki farklı boyuttaki kanalizasyon şebekesi, farklı kısıt yönetimi mekanizmaları ile çözülmüştür. Elde edilen sonuçlar, çözüm kalitesi, çözüm süresi ve gelişme oranı ölçütleri ile değerlendirilmiştir.

2 Materyal ve metod

2.1 Kanalizasyon şebekesi hidrolik tasarım optimizasyonu

Tamamen serbest yüzeyli akışa sahip bir kanalizasyon şebekesinin maliyetinin ana bileşenleri boru, muayene bacası ve kazı maliyetleridir. Kanalizasyon şebekesinin hidrolik tasarım optimizasyonu probleminde amaç, en düşük maliyete sahip şebekeyi elde etmektir. Problemin amaç fonksiyonu Denklem 1'de gösterildiği gibi tanımlanabilir [14]:

$$\min C = \sum_{i=1}^N L_i C_{p_i} + \sum_{j=1}^M C_{m_j} \quad (1)$$

Burada C kanalizasyon şebekesinin maliyetini, N toplam boru sayısı, L_i boru uzunluğunu, C_{p_i} kanalizasyon borusu inşa birim maliyetini, M toplam baca sayısı, C_{m_j} muayene bacası inşa maliyetini ifade etmektedir. C_{p_i} değeri, boru çapına (d_i), C_{m_j} ise muayene bacası derinliğine (h_{m_j}) bağlı olarak değişmektedir.

Kanalizasyon şebekelerinin projelendirilmesi sırasında diğer mühendislik projelerinde olduğu gibi uyulması gereken kurallar ve sağlanması gereken şartlar bulunmaktadır. Bu durum optimizasyon probleminde kısıtlar olarak karşımıza çıkmaktadır. Kanalizasyon şebekelerinin hidrolik tasarım optimizasyonu problemi için bu çalışmada dikkate alınan kısıtlar, Denklem 2-10 arasında verilmiştir:

$$d_i \in D, (\forall i = 1, \dots, N) \quad (2)$$

$$d_i \geq D', (\forall i = 1, \dots, N) \quad (3)$$

$$V_i \geq V_{min}, (\forall i = 1, \dots, N) \quad (4)$$

$$V_i \leq V_{max}, (\forall i = 1, \dots, N) \quad (5)$$

$$\beta_i \geq \beta_{min}, (\forall i = 1, \dots, N) \quad (6)$$

$$\beta_i \leq \beta_{max}, (\forall i = 1, \dots, N) \quad (7)$$

$$S_i \geq S_{min}, (\forall i = 1, \dots, N) \quad (8)$$

$$E_i \geq E_{min}, (\forall i = 1, \dots, N, \forall j = 1, \dots, M) \quad (9)$$

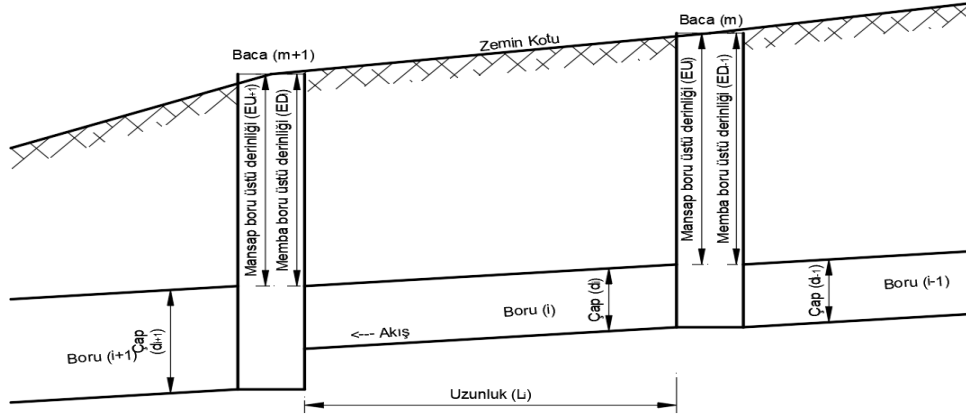
$$E_i \leq E_{max}, (\forall i = 1, \dots, N, \forall j = 1, \dots, M) \quad (10)$$

Burada D ticari boru çaplarını, D' memba boru çapını, V_i i 'inci borudaki akım hızını, V_{min} izin verilen minimum akım hızını, V_{max} izin verilen maksimum akım hızını, β_i i 'inci borudaki doluluk oranını, β_{min} izin verilen minimum doluluk oranını, β_{max} izin verilen maksimum doluluk oranını, S_i i 'inci borunun eğimini, S_{min} izin verilen minimum boru eğimini, E_i i 'inci boru üstü derinliğini, E_{min} izin verilen minimum boru üstü derinliğini, E_{max} izin verilen maksimum boru üstü derinliğini ifade etmektedir.

Hidrolik hesaplamalarda Denklem 11'de verilen Manning denklemi kullanılmıştır:

$$Q_i = \frac{1}{n} A_i R_i^{2/3} S_i^{1/2}, (\forall i = 1, \dots, N) \quad (11)$$

Burada Q_i i 'inci borudaki debiyi, n Manning katsayısını, A_i i 'inci borudaki ıslak alanı, R_i i 'inci borudaki hidrolik yarıçapı ifade etmektedir.



Şekil 1. Şebeke profili

Bahsedilen ifadeler Şekil 1'de şebeke profili üzerinde gösterilmiştir.

2.2 Karahindiba optimizasyon algoritması (Dandelion optimizer) (KOA)

Karahindiba optimizasyon algoritması Zhao vd. [28] tarafından 2022 yılında geliştirilmiş sürü zekası optimizasyon algoritmasıdır. Önerilen algoritma, karahindiba bitkisine ait tohumların rüzgârla uzun mesafe taşınması sürecinin üç aşamada matematiksel olarak modellenmesi ile geliştirilmiştir. Bu üç aşama yükselme, alçalma ve iniş olarak adlandırılmaktadır.

Başlangıçta karahindiba tohumları olarak ifade edilen ve popülasyonu oluşturan olası çözümler rastgele olarak Denklem 12 yardımı ile üretilirler. Her tohum popülasyondaki bir bireye ve dolayısıyla bir olası çözüme karşılık gelir [28].

$$x_{ij} = x_j^{min} + rand * (x_j^{max} - x_j^{min}) \quad (12)$$

Burada x_{ij} i 'nci bireyin j 'nci boyutu, x_j^{min} ve x_j^{max} sırasıyla j 'nci boyutun en küçük ve en büyük sınır değerine karşılık gelir. Başlangıç aşamasında en iyi uygunluk değeri Denklem 13 ile ifade edilmektedir. Denklem 14 ile ifade edilen en iyi uygunluk değerine sahip birey ise başlangıç elit bireyi olarak adlandırılır. Bu birey karahindiba tohumunun gelişmesi için yaklaşık en iyi konum olarak kabul edilir [28].

$$f_{best} = \min(f(X_i)) \quad (13)$$

$$X_{elite} = X(\text{find}(f_{best} == f(X_i))) \quad (14)$$

2.2.1 Yükselme aşaması

Bu aşamada karahindiba tohumları, üzerlerinde oluşan vorteks ile hava durumundan bağımsız olarak yükselir. Hava yağmurlu olduğunda ise vorteks oluşmaz ve arama yerel olur. Yükselme belirli bir değere ulaştıktan sonra tohumlar ebeveynlerinden uzaklaşabilir. Yükselme miktarı hava şartlarına (rüzgâr hızı, nem vb.) bağlıdır. Hava durumu iki şekilde ifade edilmiştir.

Birinci durum: Havanın açık olduğu bir günde rüzgâr hızının log-normal dağılıma uyduğu kabul edilebilir. Bu

kabul ile üretilen rastgele sayılar Y ekseninde daha fazla yayılmış olur, böylece tohumların daha uzağa gitme şansı artar. Bu durum karahindiba optimizasyon algoritmasının keşif aşamasıdır. Tohumların yükseleceği yükseklik rüzgâr hızı ile orantılıdır. Rüzgârdan etkilenen vorteksler tohumların spiral şeklinde yükselmesini sağlar. Bu durum Denklem 15 ile ifade edilmiştir:

$$X_{t+1} = X_t + \alpha * v_x * v_y * \ln Y * (X_s - X_t) \quad (15)$$

Burada X_t tohumun t 'inci iterasyondaki konumunu, X_s t 'inci iterasyonda arama uzayında rastgele seçilen konumu, $\ln Y \sim N(\mu, \sigma^2)$ ise log-normal dağılımı ifade etmektedir. X_s Denklem 16 ile elde edilir:

$$X_s = rand(1, Dim) * (x_j^{max} - x_j^{min}) + x_j^{min} \quad (16)$$

Denklem 17 ile hesaplanan α değeri, arama adım uzunluğunu ayarlayan uyarlanabilir bir parametredir.

$$\alpha = rand * \left(\frac{1}{T^2} t^2 - \frac{2}{T} t + 1 \right) \quad (17)$$

v_x ve v_y karahindiba kaldırma katsayısı bileşenlerine karşılık gelir ve Denklem 18-20 ile hesaplanırlar:

$$r = \frac{1}{e^\theta} \quad (18)$$

$$v_x = r * \cos \theta \quad (19)$$

$$v_y = r * \sin \theta \quad (20)$$

Burada θ $-\pi$ ile π arasında değerler alan rastgele bir sayıdır.

İkinci durum: Bu durum yağışlı bir günü ifade etmektedir. Hava şartlarından dolayı tohumlar yeterince yükselmezler ve uzaklara uçamazlar. İkinci durum Denklem 21 ile modellenmiştir:

$$X_{t+1} = X_t * k \quad (21)$$

Burada k karahindibanın yerel arama alanını düzenlemek için kullanılan bir sayı olup [Denklem 22-23](#) ile elde edilir:

$$k = 1 - rand * q \quad (22)$$

$$q = \frac{1}{T^2 - 2T + 1} t^2 - \frac{2}{T^2 - 2T + 1} t + 1 + \frac{1}{T^2 - 2T + 1} \quad (23)$$

Yükselme aşamasındaki karahindiba tohumlarının davranışının matematiksel modeli [Denklem 24](#) ile verilmiştir:

$$X_{t+1} = \begin{cases} X_t + \alpha * v_x * v_y * \ln Y * (X_s - X_t) & randn < 1.5 \\ X_t * k & \text{diğer} \end{cases} \quad (24)$$

Burada $randn$ standart normal dağılıma göre üretilen rastgele bir sayıdır [\[28\]](#).

2.2.2 Alçalma aşaması

Alçalma aşamasında, daha önce havalanan tohumlar uzayda yönlerini devamlı düzenleyerek istikrarlı bir şekilde alçalırlar. Hareket yörüngesi, Brown hareketi ile simüle edilir. Bu aşama keşif aşamasına karşılık gelir. Keşif aşaması [Denklem 25](#) ile matematiksel olarak modellenmiştir:

$$X_{t+1} = X_t - \alpha * \beta_t * (X_{mean_t} - \alpha * \beta_t * X_t) \quad (25)$$

Burada β_t Brown hareketini belirten normal dağılıma uyan rastgele bir sayıdır. X_{mean_t} i 'inci iterasyondaki popülasyon ortalama konumu olup [Denklem 26](#) ile hesaplanır. Ortalama konum bilgisi kullanılarak popülasyonun gelişimi kolaylaştırılmış olur.

$$X_{mean_t} = \frac{1}{pop} \sum_{i=1}^{pop} X_i \quad (26)$$

Burada pop popülasyon büyüklüğüdür [\[28\]](#).

2.2.3 İniş aşaması

Bu aşamada, alçalan tohumlar büyümek için rastgele konumlara inerler. Bu hareket [Denklem 27](#) ile ifade edilmiştir:

$$X_{t+1} = X_{elite} - levy(\lambda) * \alpha * (X_{elite} - X_t * \delta) \quad (27)$$

Burada X_{elite} i 'inci iterasyondaki en iyi çözümü, $levy(\lambda)$ Levy uçuşunu, ifade eder. Levy uçuşu [Denklem 28](#) ile verilmiştir:

$$Levy(\lambda) = s * \frac{\omega * \sigma}{|t|^\beta} \quad (28)$$

Burada β 0 ile 2 arasında rastgele bir sayıya, s değeri 0.001 olan sabit bir sayıya, ω ve t 0 ile 1 arasında rastgele sayıya karşılık gelir. σ ise [Denklem 29](#) ile hesaplanmaktadır:

$$\sigma = \left(\frac{\Gamma(1 + \beta) * \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) * \beta * 2^{\left(\frac{\beta-1}{2}\right)}} \right) \quad (29)$$

Burada β değeri 1.5 olan sabit bir sayıdır. δ ise [Denklem 30](#)'da verildiği gibi 0'dan 2'ye doğrusal artan bir fonksiyondur.

$$\delta = \frac{2t}{T} \quad (30)$$

Karahindiba algoritmasının akış şeması [Algoritma 1](#)'de verilmiştir [\[28\]](#).

Algoritma 1. Karahindiba algoritması [\[28\]](#)

Girdi: Birey sayısı pop , maksimum iterasyon sayısı T , karar değişkeni sayısı Dim

Çıktı: En iyi karahindiba tohumu X_{best} , ve bu tohuma ait uygunluk değeri

1. Karahindiba tohumlarının başlangıç değerlerini ata X
2. Tüm karahindiba tohumlarının uygunluk değerlerini f hesapla
3. Uygunluk değerlerine göre en iyi karahindiba tohumunu X_{elite} seç
4. **while** ($t < T$) **do**
- /*Yükselme Aşaması*/
5. **if** $randn < 1.5$ **do**
- Denklem (17) ile uyarlanabilir parametreleri oluştur
- Denklem (15) karahindiba tohumlarını güncelle
8. **else if do**
- Denklem (22-23) ile uyarlanabilir parametreleri oluştur
- Denklem (21) karahindiba tohumlarını güncelle
11. **end if**
- /*Alçalma Aşaması*/
12. Denklem (25) karahindiba tohumlarını güncelle
- /*İniş Aşaması*/
13. Denklem (27) karahindiba tohumlarını güncelle
14. Karahindiba tohumlarını uygunluk değerlerine göre iyiden kötüye doğru sırala
15. X_{elite} güncelle
16. **if** $f(X_{elite}) < f(X_{best})$
- $X_{best} = X_{elite}$; $f_{best} = f(X_{elite})$
18. **end if**
19. **end while**
20. **Return** X_{best} ve f_{best}

2.3 Kısıt-yönetimi mekanizmaları

Kısıtlı bir optimizasyon problemi [Denklem 31-33](#) gibi ifade edilebilir:

$$\min f(x) \quad (31)$$

$$g_i(x) \leq 0, i = 1, \dots, m \quad (32)$$

$$h_j(x) = 0, j = 1, \dots, p \quad (33)$$

Burada $f(x)$ maliyet fonksiyonunu, $g_i(x)$ eşitsizlik kısıtlarını, $h_j(x)$ eşitlik kısıtlarını, m eşitsizlik kısıt sayısını, p eşitlik kısıt sayısını, x çözüm vektörünü ifade etmektedir.

[Denklem 34](#), kısıtların tümünü sağlayan x kümesini, yani uygun çözüm kümesini ifade etmektedir. [Denklem 35](#) ise kısıtlardan en az bir veya birden fazlasını sağlamayan x kümesini, başka bir deyişle uygun olmayan çözüm kümesidir.

$$\mathcal{F} = \{x: g_i(x) \leq 0 \wedge h_j(x) = 0, \forall i, j\} \quad (34)$$

$$\mathcal{F} = \{x: x \notin \mathcal{F}\} \quad (35)$$

Optimizasyon yöntemlerinde ceza fonksiyonu yaklaşımı, bireyin maliyet fonksiyonunun bireyin kısıt ihlali ölçüsüne göre değiştirilmesidir. Bu durum [Denklem 36-37](#)'deki gibi ifade edilebilir:

$$\min \emptyset(x) = f(x) + \sum_{i=1}^{m+p} r_i G_i(x) \quad (36)$$

$$G_i(x) = \begin{cases} [\max(0, g_i(x))]^\beta, & \forall i \in [1, m] \\ [|\max(0, |h_i(x)| - \varepsilon)|]^\beta, & \forall i \in [m+1, m+p] \end{cases} \quad (37)$$

Burada $\emptyset(x)$ ceza terimi eklenmiş amaç fonksiyonudur [29].

2.3.1 Statik ceza yöntemi (SC)

Statik ceza yönteminde kısıt aşımına uygulanan ceza, iterasyona bağlı olmayan sabit bir değer olup [Denklem 38-39](#)'da verildiği gibi hesaplanır:

$$\min \emptyset(x) = f(x) + \sum_{i=1}^{m+p} r_i G_i(x) \quad (38)$$

$$r_i = \begin{cases} R_{i1}, & G_i(x) \in (0, T_{i1}] \\ R_{i2}, & G_i(x) \in (T_{i1}, T_{i2}] \\ \vdots & \vdots \\ R_{iq}, & G_i(x) \in (T_{i,q-1}, \infty] \end{cases} \quad (39)$$

Burada r_i ceza miktarını, $G_i(x)$ ise kısıt ihlal miktarını göstermektedir [29].

2.3.2 Dinamik ceza yöntemi (DC)

Bu yöntemde, kısıt ihlalinde uygulanan ceza miktarı iterasyon ilerledikçe arttığından dinamik ceza yöntemi adını almıştır. Dinamik ceza yönteminin hesaplanması [Denklem 40-41](#)'de verilmiştir:

$$\min \emptyset(x) = f(x) + (ct)^\alpha M(x) \quad (40)$$

$$M(x) = \sum_{i=1}^{m+p} G_i(x) \quad (41)$$

Burada c ve α ceza katsayılarını, t ilgili iterasyon adımını, $M(x)$ toplam kısıt ihlal miktarını göstermektedir [29].

2.3.3 Olurlu çözümlerin üstünlüğü ile birleştirilmiş dinamik ceza yöntemi (BDC)

Dinamik ceza yöntemindeki bir diğer seçenekte, dinamik ceza yöntemini olurlu çözümlerin üstünlüğü yöntemiyle birleştirmektedir. Bu yöntem dinamik ceza yöntemine ek olarak, kısıt ihlali yapan çözümlere $\theta(x)$ fonksiyonuyla hesaplanan ek bir ceza uygulanmasıdır. Bu yöntemde kullanılan $\theta(x)$ fonksiyonu, tüm olurlu çözümlerin olurlu olmayan çözümlere göre daha iyi bir uygunluk değerine sahip olmalarını sağlar. BDC yönteminin hesaplanması [Denklem 42-44](#)'te verildiği gibidir [29]:

$$\emptyset(x) = \begin{cases} f(x), & x \in \mathcal{F} \\ f(x) + (ct)^\alpha M(x) + \theta(x), & x \notin \mathcal{F} \end{cases} \quad (42)$$

$$M(x) = \sum_{i=1}^{m+p} G_i(x) \quad (43)$$

$$\theta(x) = \quad (44)$$

$$\begin{cases} 0, & x \in \mathcal{F} \\ \max[0, \max_{y \in \mathcal{F}} f(y) - \min_{y \notin \mathcal{F}} \emptyset(y)], & x \notin \mathcal{F} \end{cases}$$

2.3.4 Üstel dinamik ceza yöntemi (ÜDC)

Üstel dinamik ceza yönteminde, dinamik ceza yönteminde olduğu gibi ceza miktarı iterasyonla artmaktadır. T değeri, iterasyon sayısı sonsuza yaklaştıkça olurlu çözüm sağlamayan bireylerin uygunluk değerlerinin de sonsuza yakınsamasını sağlar. ÜDC yönteminin hesaplanması [Denklem 45-46](#)'da verilmiştir:

$$\min \emptyset(x) = f(x)e^{(M(x)/T)} \quad (45)$$

$$T = 1/\sqrt{t} \quad (46)$$

Burada $M(x)$ toplam kısıt ihlal miktarını göstermektedir [29].

2.3.5 Olurlu çözümlerin üstünlüğü ile birleştirilmiş üstel dinamik ceza yöntemi (BÜDC)

Bu yöntemde üstel dinamik ceza yöntemi ile olurlu çözümlerin üstünlüğü yöntemi birleştirilmiştir. Üstel dinamik ceza yöntemine ek olarak kısıt ihlali yapan çözümlere $\theta(x)$ fonksiyonuyla hesaplanan ek bir ceza uygulanmaktadır. Bu yöntemde kullanılan $\theta(x)$ fonksiyonu, tüm olurlu çözümlerin olurlu olmayan çözümlere göre daha iyi bir uygunluk değerine sahip olmalarını sağlar. BÜDC yönteminin hesaplanması [Denklem 47-48](#)'de gösterilmiştir [29]:

$$\emptyset(x) = \begin{cases} f(x), & x \in \mathcal{F} \\ f(x)e^{(M(x)/T)} + \theta(x), & x \notin \mathcal{F} \end{cases} \quad (47)$$

$$= \begin{cases} \theta(x) & \\ \max[0, \max_{y \in \mathcal{F}} f(y) - \min_{y \notin \mathcal{F}} \emptyset(y)], & x \notin \mathcal{F} \end{cases} \quad (48)$$

2.3.6 Eklektik ceza yöntemi (EC)

Eklektik Ceza yöntemi olurlu çözümlerin üstünlüğünü sağlamak için [Denklem 49](#)'da verildiği üzere başka bir yaklaşım önerir:

$$\emptyset(x) = \begin{cases} f(x), & x \in \mathcal{F} \\ K \left(1 - \frac{s(x)}{m+p}\right), & x \notin \mathcal{F} \end{cases} \quad (49)$$

Burada K olurlu çözümlerin uygunluk değerinin olurlu olmayan çözümlerin uygunluk değerinden iyi olmasını sağlayan bir katsayıyı, $s(x)$ sağlanan kısıt sayısını, $m+p$ ise toplam kısıt sayısını göstermektedir [29].

2.3.7 Uyarlanabilir ceza yöntemi (UC)

Uyarlanabilir ceza yönteminde ceza üç farklı duruma göre [Denklem 50](#)'de verildiği şekli ile hesaplanmaktadır:

$$r_i(t+1) = \begin{cases} \frac{r_i(t)}{\beta_1}, & \text{Durum 1} \\ \beta_2 r_i(t), & \text{Durum 2} \\ r_i(t), & \text{diğer} \end{cases} \quad (50)$$

Burada *Durum 1* geçmiş k iterasyonun her birinde en iyi bireyin olurlu olduğu, *Durum 2* geçmiş k iterasyonun hiçbirinde olurlu birey olmadığı, *diğer* ise bu iki durum dışındaki durumları ifade etmektedir. Genellikle $r_i(1)=1$, $\beta_1=4$, $\beta_2=3$ alınırken, k değeri problem boyutuna eşit olarak alınır [29].

2.3.8 Ters tanjant yöntemi (TT)

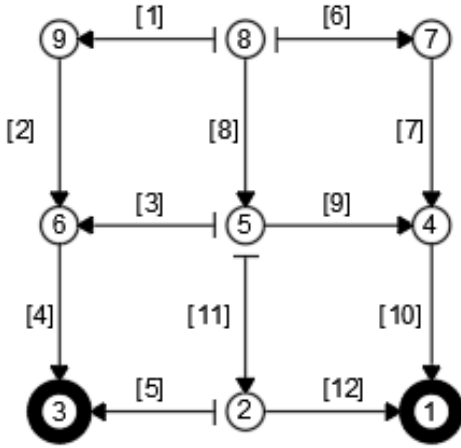
İterasyondaki bireylerin kısıtları ihlal etmesi durumunda, ihlal edilen kısıtların en büyük değeri ($g_{max}(\vec{x})$) uygunluk değeri olarak belirlenir. Öte yandan ihlal edilen kısıt bulunmaması durumunda ise, uygunluk değeri amaç fonksiyonunun ters tanjant değeri olarak alınır. Ters tanjant yönteminde uygunluk değeri Denklem 51'de verildiği gibi hesaplanır:

$$f(\vec{x}) = \begin{cases} \hat{g}(\vec{x}) = g_{max}(\vec{x}), & g_{max}(\vec{x}) > 0 \\ \hat{f}(\vec{x}) = atan[f(\vec{x})] - \pi/2, & \text{diğer} \end{cases} \quad (51)$$

Burada $atan[.]$ ters tanjant fonksiyonunu ifade etmektedir [30].

2.4 Uygulama

Çalışma için, literatürde bulunan iki farklı büyüklükteki şebeke seçilmiştir. Birinci şebeke, ilk olarak Moeini ve Afshar [31] tarafından önerilen ve daha sonraki çalışmalarda farklı planlara sahip olarak çalışılmış bir şebekedir. Bu çalışmada Şekil 2'de gösterildiği gibi, şebekenin Moeini ve Afshar [32] tarafından önerilen hali kullanılmaktadır.



Şekil 2. Birinci şebeke [32]

Bu şebekede 9 baca ve her birinin uzunluğu 100 m- olan 12 hat bulunmaktadır. Başlangıç bacası 8'in zemin kotu 1000 m-'dir. Diğer bacaların zemin kotları, akış yönünde %2 azalacak şekilde hesaplanmaktadır. Şebekede çözüm için $V_{min}=0.75$ m/s, $V_{max}=6$ m/s, $\beta_{min}=0.1$, $\beta_{max}=0.83$, $E_{min}=2.5$ m, $E_{max}=10$ m, $S_{min}=0.0005$, $n=0.013$, D (mm)=[100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 660, 650, 700, 750, 800, 850, 900, 950, 1000, 1100, 1200, 1300, 1400, 1500] olarak alınmıştır. Şebeke için maliyet fonksiyonu Denklem 52 ve 53'te verilmektedir [32]:

$$K_p = 10.93e^{3.43d_i} + 0.012\bar{E}_i^{1.53} + 0.437\bar{E}_i^{1.47}d_i \quad (52)$$

$$K_m = 41.46h_{m_i} \quad (53)$$

Burada i dikkate alınan boruyu, K_p kanalizasyon borusu inşa birim maliyetini, d_i boru çapını, \bar{E}_i ortalama boru üstü derinliğini, K_m baca inşa maliyetini, h_{m_i} baca derinliğini ifade etmektedir.

Çalışmada kullanılan ikinci şebeke, ilk olarak Mansouri ve Khanjani [33] tarafından önerilen ve Şekil 3'te verilen Kerman Şehrine ait şebekedir. Bu şebekede 21 baca ve 20 hat bulunmaktadır. Şebekede çözüm için $V_{min}=0.3$ m/s, $V_{max}=3$ m/s, $\beta_{min}=0.1$, $\beta_{max}=0.82$, $E_{min}=2.45$ m, $n=0.013$, D (mm)=[150, 200, 250, 300, 350, 400, 450, 500, 600, 700] olarak alınmıştır [33]. Hassan vd.'nin [34] çalışmasında belirtildiği gibi bu şebeke ile yapılan bazı çalışmalarda kazı derinliği olarak boru altı derinliği dikkate alınırken, bazı çalışmalarda ise boru üstü derinliği dikkate alınmıştır. Bu durum çalışmalarda maliyet değerlerinin farklı olarak elde edilmesine neden olmaktadır. Bu çalışmada ise, kazı derinliği olarak boru üstü derinliği dikkate alınmıştır. Şebeke için maliyet fonksiyonu Denklem 54 ve 55'te verilmektedir [33]:

$$K_p = 1.93e^{3.43d_i} + 0.812\bar{E}_i^{1.53} + 0.437\bar{E}_i^{1.47}d_i \quad (54)$$

$$K_m = 41.46h_{m_i} \quad (55)$$

Denklem 52-53 ve 54-55'te verilen boru ve baca inşa maliyet değerlerinin toplamı, Denklem 56'da verilen toplam şebeke maliyetini K_T ifade etmektedir:

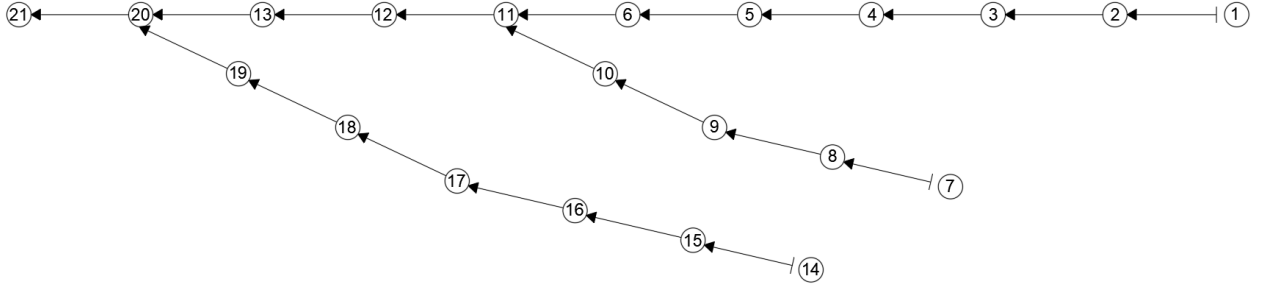
$$K_T = K_p + K_m \quad (56)$$

Çalışmada kullanılan karar değişkenleri, boru çapları ve her borunun memba ve mansabındaki boru üstü derinlikleridir. Birinci şebekede 12 hat, her hat için 3 değişken olmak üzere toplam 36 karar değişkeni, ikinci şebekede ise 20 hat, her hat için 3 değişken olmak üzere 60 karar değişkeni bulunmaktadır.

Çalışmada kullanılan KOA'nın birey sayısı ve iterasyon sayısı olmak üzere 2 parametresi bulunmaktadır. Birey sayısı birinci şebeke için 50, ikinci şebeke için 100 olarak alınmıştır. İterasyon sayısı ise her iki şebeke içinde 40000 olarak belirlenmiştir. Sonuçları istatistiksel açıdan değerlendirebilmek amacıyla her bir farklı kısıt mekanizması için 30 bağımsız koşum yapılmıştır. Her bir koşum için başlangıç popülasyonunun kısıtları sağlanmasına dikkat edilmiştir. Bir başka deyişle başlangıç popülasyonu olurlu çözümlerden meydana gelmektedir.

3 Bulgular ve tartışma

Kısıt yönetimi mekanizmalarının çözüm kalitesi ve süresi üzerindeki etkisini ölçmek için KOA, her iki şebekede de her bir kısıt yönetimi mekanizması için 30 defa koşturulmuştur. Bu 30 koşumda elde edilen sonuçların en iyi, ortalama, en kötü değerleri ve standart sapma değerleri kaydedilmiştir. Bu veriler Tablo 1'de özetlenmiştir.



Şekil 3. İkinci şebeke [33]

Tablo 1. İstatistiksel sonuçlar

Şebeke	Performans Ölçütü	Kısıt Yönetimi Mekanizmaları							
		SC	DC	BDC	ÜDC	BÜDC	EC	UC	TT
Şebeke 1	En iyi	23001.66	23001.98	23233.03	23001.87	23004.12	23006.31	23035.61	23233.03
	Ortalama	23504.13	23521.28	23502.72	23432	23372.76	23512.24	23343.09	23420.13
	En Kötü	24602.52	24846.31	24498.32	24524.03	24614.8	24565.08	24013.99	24996.72
	Standart Sapma	415.07	486.72	356.94	355.43	332.83	373.41	189.99	339.65
Şebeke 2	En iyi	82897.51	82384.18	82324.29	82712.34	82580.93	82638.90	83263.91	82603.49
	Ortalama	87646.46	87027.18	88551.82	88294.21	87372.69	86964.09	85678.31	86485.10
	En Kötü	102362.39	97716.64	109007.33	104903.72	102510.57	100317.88	91288.97	97896.73
	Standart Sapma	4246.38	2813.12	5439.35	5022.66	4086.42	3656.28	2116.35	3663.77

Tablo 1'den de görüldüğü gibi şebeke 1 için uygunluk fonksiyonu değerinin en iyi sonuç olduğu kısıt yönetimi mekanizması, statik ceza (SC) yöntemidir. Öte yandan, uygunluk değerleri ortalama ve standart sapma performans ölçütleri açısından değerlendirildiğinde ise uyarlamalı ceza (UC) yönteminin diğer mekanizmalara göre daha iyi sonuca sahip olduğu görülmüştür.

Tablo 1'e göre şebeke 2 için sonuçlar değerlendirildiğinde en iyi sonuca olurlu çözümlerin üstünlüğüyle birleştirilmiş dinamik ceza (BDC) yöntemi sahiptir. Bunun yanında, uyarlanmış ceza (UC) yöntemi şebeke 1'de olduğu gibi şebeke 2'de de ortalama ve standart sapma ölçütlerine göre en iyi sonuca sahiptir.

Kısıt yönetimi mekanizmalarını ortalama performans değerlerine göre karşılaştırmak için Friedman Testi uygulanmıştır. Friedman testi sonuçları Tablo 2'de verilmiştir. Sonuçlara göre en başarılı mekanizma uyarlamalı ceza (UC) yöntemi olurken, ters tanjant (TT) yöntemi ikinci sırada yer almıştır.

Tablo 2. Friedman test sonuçları

Kısıt Yönetimi Mekanizması	Friedman Test Ortalaması (Sıralama)
SC	4.68 (5)
DC	4.62 (4)
BDC	5.07 (8)
ÜDC	4.77 (7)
BÜDC	4.30 (3)
EC	4.68 (5)
UC	3.75 (1)
TT	4.13 (2)

Sonuçlar ayrıca gelişim oranı (progress ratio) açısından da değerlendirilmiştir. Gelişim oranı hesaplamadaki amaç, algoritmanın arama uzayının uygulanabilir bölgesi içindeki

iyileştirme kabiliyetini ölçmektir. Bu ölçüm için yüksek değerler tercih edilir, çünkü bunlar daha yüksek bir gelişmeyi gösterirler. Gelişim oranı Denklem 57 ile hesaplanır:

$$\text{Gelişim oranı} = \begin{cases} \ln \left| \frac{f(X_{ilk})}{f(X_{eni})} \right| & \text{eğer, } f(X_{eni}) > 0 \\ \ln \left| \frac{f(X_{ilk}) + 1}{f(X_{eni}) + 1} \right| & \text{eğer, } f(X_{eni}) = 0 \\ \ln \left| \frac{f(X_{ilk}) + 2|f(X_{eni})|}{f(X_{eni}) + 2|f(X_{eni})|} \right| & \text{eğer, } f(X_{eni}) < 0 \end{cases} \quad (57)$$

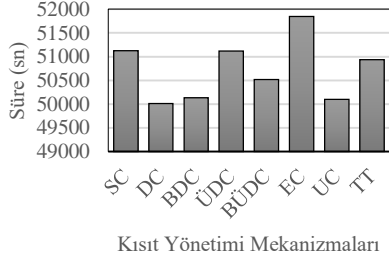
Kısıt yönetimi mekanizmalarına ait gelişim oranı değerleri Tablo 3'te verilmiştir. Gelişim oranına göre en iyi mekanizma eklettik ceza (EC) yöntemi olurken, ters tanjant (TT) yöntemi ikinci sırada yer almıştır.

Tablo 3. Gelişim oranı katsayısı test sonuçları

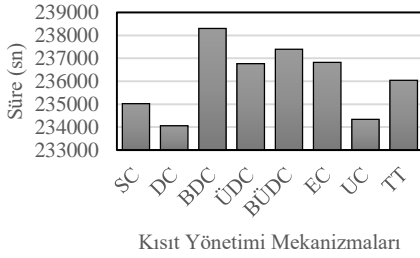
Kısıt Yönetimi Mekanizması	Gelişim Oranı
SC	0.4405 (5)
DC	0.4352 (8)
BDC	0.4364 (7)
ÜDC	0.4405 (5)
BÜDC	0.4417 (3)
EC	0.4474 (1)
UC	0.4416 (4)
TT	0.4446 (2)

Her kısıt yönetimi mekanizması çalışma süreleri açısından karşılaştırılmıştır. Her bir mekanizma için çalışma

süresi, 30 koşumun sürelerinin ortalamasıdır. Çalışma sürelerinin çubuk grafikleri Şekil 4 ve 5'te verilmiştir. Çalışma süreleri açısından mekanizmalar arasında çok fark olmamasına rağmen, uyarlamalı ceza (UC) yöntemi ve dinamik ceza (DC) yöntemi en kısa çalışma süresine sahiptir.



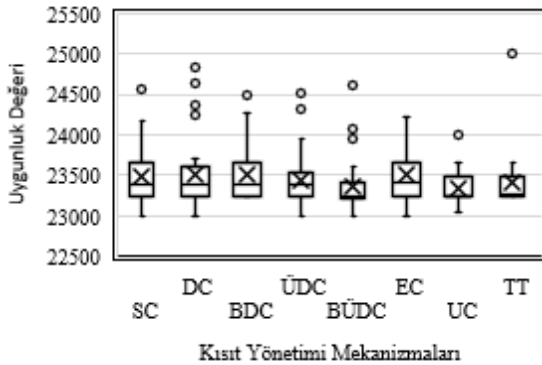
Şekil 4. Şebeke 1 için çalışma süreleri



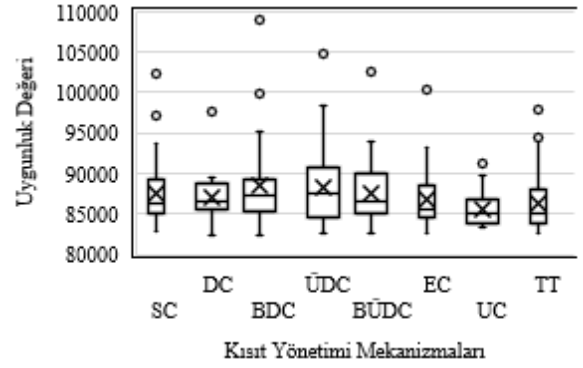
Şekil 5. Şebeke 2 için çalışma süreleri

Son olarak, kısıt yönetimi mekanizmalarına ait sonuçlar iki şebeke için ayrı ayrı kutu grafikler olarak şebeke 1 için Şekil 6'da, şebeke 2 için Şekil 7'de verilmiştir. Şekil 6'ya göre olurlu çözümlerin üstünlüğü ile birleştirilmiş dinamik ceza (BDC) yöntemi en az yayılıma sahip kutu grafiğine sahiptir. Bunun yanında uyarlamalı ceza (UC) yöntemine ait medyan değeri diğer mekanizmalardan küçüktür. Bu da uyarlamalı ceza (UC) yönteminin en başarılı mekanizma olduğunu göstermektedir.

Şekil 7'ye göre en az yayılıma sahip kutu grafiğine sahip mekanizma uyarlamalı ceza (UC) yöntemidir. Bunun yanında uyarlamalı ceza (UC) yöntemine ait medyan değeri diğer mekanizmalardan küçüktür.



Şekil 6. Şebeke 1 kutu grafiği



Şekil 7. Şebeke 2 kutu grafiği

4 Sonuçlar

Bu çalışmada statik ceza yöntemi, dinamik ceza yöntemi, olurlu çözümlerin üstünlüğü ile birleştirilmiş dinamik ceza yöntemi, üstel dinamik ceza yöntemi, olurlu çözümlerin üstünlüğü ile birleştirilmiş üstel dinamik ceza yöntemi, eklettik ceza yöntemi, uyarlanabilir ceza yöntemi ve ters tanjant yöntemleri kullanılarak kısıt yönetimi mekanizmalarının kanalizasyon şebekesi hidrolik tasarım optimizasyonu probleminin çözümünde performansa etkisi araştırılmıştır. Değerlendirme için iki farklı boyutta kanalizasyon şebekesi kullanılarak sonuçlar elde edilmiştir. Performans ölçütü olarak çözüm kalitesi, çözüm süresi ve gelişme oranı ölçütleri kullanılmıştır.

Karahindiba Optimizasyon Algoritması seçilerek bu problemin çözümü için literatürde ilk defa kullanılmıştır.

Elde edilen sonuçlara göre her ne kadar birinci şebekede en iyi sonucu statik ceza yöntemi, ikinci şebekede ise en iyi sonucu olurlu çözümlerin üstünlüğü ile birleştirilmiş dinamik ceza yöntemi vermiş olsa da sonuçlar ortalama ve standart sapmaya göre değerlendirildiğinde uyarlamalı ceza yönteminin her iki şebekede de en uygun sonuca sahip olduğu görülmüştür. Metasezgisel algoritmalar yapıları gereği zaman zaman iyi sonuçlar verse de önemli olan nokta, tutarlı sonuçlara sahip olmalarıdır. Sonuçlar bu açıdan değerlendirildiğinde uyarlamalı ceza yönteminin diğer kısıt yönetimi mekanizmalarının önünde olduğu görülmektedir. Sonuçlar gelişim oranı açısından değerlendirildiğinde, eklettik ceza yönteminin gelişim oranı en fazla olan mekanizma olduğu anlaşılmaktadır. Sonuçlar çalışma süresi açısından değerlendirildiğinde ise dinamik ceza yönteminin en kısa çalışma süresine sahip olduğu, en tutarlı sonuçlara sahip olan uyarlamalı ceza yönteminin çalışma süresi açısından ikinci sırada yer aldığı görülmektedir. Friedman test sonuçlarına göre uyarlamalı ceza yöntemi ilk sırada, ters tanjant yöntemi ise ikinci sıradadır.

Bu bilgiler ışığında uyarlamalı ceza yönteminin seçilebilecek en uygun kısıt yönetimi mekanizması olduğu söylenebilir. Fakat bu yöntem yapısı gereği parametre optimizasyonu gerektirdiği için, parametresi olmayan ters tanjant yöntemi de bir alternatif olarak seçilebilir.

Çıkar çatışması

Yazarlar çıkar çatışması olmadığını beyan etmektedir.

Benzerlik oranı (iThenticate): %8

Kaynaklar

- [1] L. W. Mays and B. C. Yen, Optimal cost design of branched sewer systems. *Water Resources Research*, 12(1), 37–47, 1975.
- [2] L. W. Mays and H. G. Wenzel, Optimal design of multilevel branching sewer systems. *Water Resources Research*, 12(5), 913–917, 1976.
- [3] L. Y. Liang, R. G. Tompson, and D. M. Young, Optimising the design of sewer networks using genetic algorithms and tabu search. *Engineering, Construction and Architectural Management*, 11(2), 101–112, 2004. <https://doi.org/10.1108/09699980410527849>.
- [4] M. H. Afshar, A. Afshar, M. A. Mariño, and A. A. S. Darbandi, Hydrograph-based storm sewer design optimization by genetic algorithm. *Canadian Journal of Civil Engineering*, 33(3), 319–325, 2006. <https://doi.org/10.1139/L05-121>.
- [5] T.-C. Pan and J.-J. Kao, GA-QP model to optimize sewer system design. *Journal of Environmental Engineering*, 135(1), 17–24, 2009. [https://doi.org/10.1061/\(ASCE\)07339372\(2009\)135:1\(17\)](https://doi.org/10.1061/(ASCE)07339372(2009)135:1(17)).
- [6] M. H. Afshar, A parameter free continuous ant colony optimization algorithm for the optimal design of storm sewer networks: constrained and unconstrained approach. *Advances in Engineering Software*, 41(2), 188–195, 2010. <https://doi.org/10.1016/j.advengsoft.2009.09.009>.
- [7] T. Cetin and M. A. Yurdusev, Genetic algorithm for networks with dynamic mutation rate. *Gradevinar*, 69(12), 1101–1109, 2018. <https://doi.org/10.14256/JCE.1533.2015>.
- [8] Masoumi, F., S. Masoumzadeh, N. Zafari, and M.J.E. Skardi, Optimum sanitary sewer network design using shuffled gray wolf optimizer. *Journal of Pipeline Systems Engineering and Practice* 12(4): 04021055, 2021. [https://doi.org/10.1061/\(ASCE\)PS.1949-1204.0000597](https://doi.org/10.1061/(ASCE)PS.1949-1204.0000597).
- [9] A. Gholami, P. G. Durgut and M. T. Ayvaz, An integrated simulation-optimization approach for dynamic design of the urban wastewater collection systems. *Turkish Journal of Civil Engineering*, 2023 105-134, 719, 2023. <https://doi.org/10.18400/tjce.1209180>.
- [10] K. N. Praveen and Y. P. Mathur, Application of graph theory for optimal sewer layout generation. *Discovery*, 40(183), 151-157, 2015.
- [11] A. E. Bakhshipour, M. Bakhshizadeh, U. Dittmer, W. Nowak, and A. Haghighi, A graph-theory based algorithm to generate decentralized urban drainage layouts. *Green*, 633–637, 2018. https://doi.org/10.1007/978-3-319-99867-1_109.
- [12] N. de Villiers, G. C. van Rooyen and M. Middendorf, Sewer network design layout optimisation using ant colony algorithms. *Journal of the South African Institution of Civil Engineers*, 60(3), September 2018, 2–15, 1773, 2018. <https://doi.org/10.17159/2309-8775/2018/v60n3a1>.
- [13] G. P. W. Rodrigues, L. H. M. Costa, G. M. Farias and M. A. H. de Castro, A depth-first search algorithm for optimizing the gravity pipe networks layout. *Water Resources Management*, 33,4583–4598, 2019. <https://doi.org/10.1007/s11269-019-02373-x>.
- [14] M. E. Turan, G. Bacak-Turan, T. Cetin and E. Aslan, Feasible sanitary sewer network generation using graph theory. *Advances in Civil Engineering*, 2019, 8527180, 15, 2019. <https://doi.org/10.1155/2019/8527180>.
- [15] G. Li and R. G. S. Matthew, New approach for optimization of urban drainage systems. *Journal of Environmental Engineering*, 116(5),927–944, 1990. [https://doi.org/10.1061/\(ASCE\)07339372\(1990\)116:5\(927\)](https://doi.org/10.1061/(ASCE)07339372(1990)116:5(927)).
- [16] A. F. Diogo and V. M. Graveto, Optimal layout of sewer systems: a deterministic versus a stochastic model. *Journal of Hydraulic Engineering*, 132(9), 927–943, 2006. [https://doi.org/10.1061/\(ASCE\)0733-9429\(2006\)132:9\(927\)](https://doi.org/10.1061/(ASCE)0733-9429(2006)132:9(927)).
- [17] N. Duque, D. Duque, A. Aguilar and J. Saldarriaga, Sewer network layout selection and hydraulic design using a mathematical optimization framework. *Water*, 12, 3337, 2020. <https://doi.org/10.3390/w12123337>.
- [18] F. M. Alfaisal and L. W. Mays, Optimization models for layout and pipe design for storm sewer systems. *Water Resources Management*, 35,4841–4854, 2021. <https://doi.org/10.1007/s11269-021-02958-5>.
- [19] E. Tan, D. Sadak, and M.T. Ayvaz, Kanalizasyon sistemlerinin diferansiyel evrim algoritması kullanılarak optimum tasarımı. *Teknik Dergi*, 31(5), 10229-10250, 2020. <https://doi.org/10.18400/tekderg.541507>.
- [20] M. Çunkaş, and A. Ürkmez, Çok kriterli bulanık genetik algoritma ile dalgıç asenkron motorların tasarım optimizasyonu. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 23(3), 2008.
- [21] Z. Garip, M.E. Çimen, and A. F. Boz, Harris şahinleri ve balina optimizasyon algoritmalarının kısıt işleme teknikleriyle uygulaması: Karşılaştırmalı bir çalışma. *Journal of Intelligent Systems: Theory and Applications*, 4(2), 76-85, 2021. <https://doi.org/10.38016/jista.857881>.
- [22] A. Baykasoğlu, and F.B. Ozsoydan, Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Applied soft computing*, 36, 152-164, 2015. <https://doi.org/10.1016/j.asoc.2015.06.056>.
- [23] M.F. Tasgetiren, A Genetic Algorithm with an Adaptive Penalty Function for the Orienteering Problem. *Journal of Economic & Social Research*, 4(2), 2002.
- [24] G. Iyengar, and K. Sigman, Exponential penalty function control of loss networks. *The Annals of Applied Probability*, 14(4), 1698–1740, 2004. <https://doi.org/10.1214/10505160400000936>.

- [25] İ. Gölcük, A comparative analysis of constraint-handling mechanisms for solving engineering design problems. *Endüstri Mühendisliği*, 32(2), 201-216, 2021. <http://orcid.org/0000-0002-8430-7952>.
- [26] O. Kramer, A review of constraint-handling techniques for evolution strategies. *Applied Computational Intelligence and Soft Computing*, 1-19, 2010. <https://doi.org/10.1155/2010/185063>.
- [27] M. Y. Ameca-Alducin, M. Hasani-Shoreh, W. Blaikie, F. Neumann and E. Mezura-Montes, A comparison of constraint handling techniques for dynamic constrained optimization problems. 2018 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE, 2018. <https://doi.org/10.1109/CEC.2018.8477750>.
- [28] S. Zhao, T. Zhang, S. Ma and M. Chen, Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications. *Engineering Applications of Artificial Intelligence*, 114, 105075, 2022. <https://doi.org/10.1016/j.engappai.2022.105075>.
- [29] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.
- [30] T. H. Kim, I. Maruta, and T. Sugie, A simple and efficient constrained particle swarm optimization and its application to engineering design problems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 224(2), 389-400, 2010. <https://doi.org/10.1243/09544062JMES1732>.
- [31] R. Moeini and M. H. Afshar, Layout and size optimization of sanitary sewer network using intelligent ants. *Advances in Engineering Software* 51,49–62, 2012. <https://doi.org/10.1016/j.advengsoft.2012.05.003>.
- [32] R. Moeini and M. H. Afshar, Arc based ant colony optimization algorithm for optimal design of gravitational sewer networks. *Ain Shams Engineering Journal* 8, 207–223, 2017. <https://doi.org/10.1016/j.asej.2016.03.003>.
- [33] M. Mansouri, M. Khanjani, Optimization of sewer networks using nonlinear programming. *J Water Wastewater*, 10(2), 20-30,1999.
- [34] W. H. Hassan, M. H. Jassem, and S. S. Mohammed, A GA-HP model for the optimal design of sewer networks. *Water Resources Management* 32:865-879, 2018. <https://doi.org/10.1007/s11269-017-1843-y>.

