

Meta-Sezgisel Optimizasyon Yöntemlerinin Karşılaştırılması: Parçacık Sürüsü, Diferansiyel Evrim, Benzetilmiş Tavlama ve Tabu Arama

Zeliha Ilgın Güven (032390013)
Ömer Faruk Arı(032390057)

Kasım 2025

1 Giriş

Optimizasyon, hem doğal yasalar hem de insan faaliyetleri bağlamında en iyi sonucu elde etme çabasını temsil eden temel bir kavramdır. En geniş anlamıyla, belirli bir durumda en uygun (optimal) çözümü elde etmek amacıyla bir sistemin çıktısını en aza indirme veya en üste çıkarma süreci olarak tanımlanabilir. Günümüzde kaynakların giderek daha kıt hale geldiği bir dünyada, mevcut kaynakların—zaman, enerji, iş gücü, malzeme veya sermaye gibi—en verimli şekilde kullanılması gerekliliği, optimizasyonun önemini daha da vurgulamaktadır. Bu bağlamda, optimizasyon yalnızca matematiksel bir süreç değil, aynı zamanda sınırlı girdilerden maksimum çıktıyı elde etmeyi amaçlayan evrensel bir karar alma yaklaşımıdır. Dahası, literatür, optimizasyonun çeşitli alanlarda geniş bir uygulama yelpazesine sahip olduğunu ve bu alanda geliştirilen algoritmaların birçok problem alanında başarılı sonuçlar ürettiğini ortaya koymaktadır (Yang, 2012).

Optimizasyon problemleri, mühendislikten kimyaya kadar geniş bir yelpazede ortaya çıkar. Örneğin, analitik kimyada, sınırlı sayıda deneyle maksimum bilgi elde etmek için en uygun deneysel koşulları veya cihaz ayarlarını belirlemek, klasik bir optimizasyon problemi örneğidir (Wehrens Buydens, 2000). Bu tür durumlarda, yanıt yüzey modelleri veya simplex optimizasyon teknikleri sıklıkla kullanılır; ancak bu yöntemler yalnızca belirli varsayımların sağlanması durumunda etkili sonuçlar verir (Wehrens Buydens, 2000). Bu durum, yalnızca klasik optimizasyon yaklaşımlarının güçlü yönlerini değil, aynı zamanda karmaşık ve doğrusal olmayan problemlerle başa çıkmadaki sınırlamalarını da göstermektedir.

Buna rağmen, klasik optimizasyon yöntemleri—lineer programlama, gradyan tabanlı veya deterministik algoritmalar gibi—genellikle temel sınırlamalarla karşı karşıyadır; bunlar arasında yerel optimuma takılma, global optimumu bulamama ve yüksek boyutlu problemler için düşük verimlilik yer alır. Özellikle, gradyan tabanlı algoritmalar, karmaşık, çok modlu hedef fonksiyonlarına uygulandığında yerel minimumlarda takılabilir ve böylece global çözüme ulaşamaz

(Wang Chen, 2013). Ayrıca, türev bilgisine bağımlılıkları, bu yöntemlerin kesikli veya gürültülü fonksiyonlara uygulanabilirliğini sınırlar.

Bu tür sınırlamaların üstesinden gelmek için, stokastik bileşenler içeren esnek arama stratejilerine sahip meta-sezgisel yaklaşımlar geliştirilmiştir. Bir meta-sezgisel algoritma, arama alanını keşfetmek ve kullanmak amacıyla farklı kavramları akılcıca birleştiren yinelemeli bir üretim süreci olarak geniş anlamda tanımlanabilir ve böylece alt seviye sezgisel yöntemleri yönlendirir (Said, Mahmoud, El-Horbaty, 2014). Bu süreçte, öğrenme stratejileri, neredeyse-optimal çözümleri verimli bir şekilde keşfetmeyi sağlayacak bilgi yapıları organize eder. Meta-sezgisel algoritmaların temelindeki yinelemeli hesaplama mekanizması, aday çözümleri belirli bir kalite kriterine göre ardışık olarak iyileştirir (Wang Chen, 2013).

Sonuç olarak, meta-sezgisel algoritmalar, stokastiklik sayesinde yerel minimumlardan kaçabilir, global aramalar yapabilir ve büyük ve karmaşık arama alanlarında neredeyse-optimal çözümleri verimli bir şekilde üretebilir; bu nedenle klasik yöntemlere güçlü bir alternatif sunar.

Bu bağlamda, Particle Swarm Optimization (PSO), Differential Evolution (DE), Simulated Annealing (SA) ve Tabu Search (TS), mühendislik, jeofizik (Balkaya, Göktürkler, Ekinci, Karaoğlu, 2014) ve veri bilimi (Cheng ve ark., 2016) gibi alanlarda kullanılan önde gelen meta-sezgisel yöntemler olarak öne çıkmaktadır. Doğal süreçlerden veya bilişsel davranışlardan esinlenen bu algoritmalar, karmaşık çözüm alanlarında keşif ve kullanımı dengeleyerek etkili çözümler üretir.

Bu rapor, literatür odaklı bir yaklaşım temelinde, bu dört meta-sezgisel yöntemi yapısal özellikleri, arama mekanizmaları ve uygulama alanları açısından detaylı olarak analiz etmeyi amaçlamaktadır. Analiz, her algoritmanın hangi koşullar altında, neden ve nasıl etkili olduğunu ortaya koymayı, bu yöntemlerin optimizasyon problemlerini çözmedeki uygun kullanım senaryolarını ve karşılaştırmalı avantajlarını açıklamayı hedeflemektedir.

Tüm kod örnekleri, algoritma simülasyonları ve figür üretim betikleri, raporun orijinal versiyonu ile birlikte GitHub deposunda okunmaya açıktır: Github: <https://github.com/zelihaguven/Optimization-Algorithms-Report>

2 Yaklaşım

Bu bölümde, raporda analiz edilen dört meta-sezgisel algoritma: PSO, DE, SA ve TS ele alınmaktadır.

2.1 Particle Swarm Optimization (PSO)

Parçacık Sürüsü Optimizasyonu, kuş sürülerinin veya balık gruplarının sosyal davranışlarından esinlenen popülasyon tabanlı bir meta-sezgisel optimizasyon algoritmasıdır (Naser Naser, 2024) (Bolufé-Röhler Tamayo-Vera, 2025). 1995 yılında Kennedy ve Eberhart tarafından önerilen PSO, her biri aday çözüm olarak adlandırılan bir parçacık sürüsünden oluşur; her bir parçacık, kendi en

iyi bulduğu pozisyon ve sürüdeki herhangi bir parçacığın bulduğu global en iyi pozisyon temelinde iteratif olarak hareket yolunu ayarlar (Nasr et al., 2020). Bu işbirlikçi arama mekanizması, sürünün arama alanını topluca keşfetmesini ve optimal çözümlere yakınsamasını sağlar (Okechukwu et al., 2024). Bu yaklaşım, PSO'yu diğer meta-sezgisel yöntemlerden ayırır; çünkü PSO, karmaşık genetik operatörler veya simüle edilmiş fiziksel süreçler yerine ortaya çıkan kolektif zekâyâ dayanır (Cuevas et al., 2024) (Martinez et al., 2020).

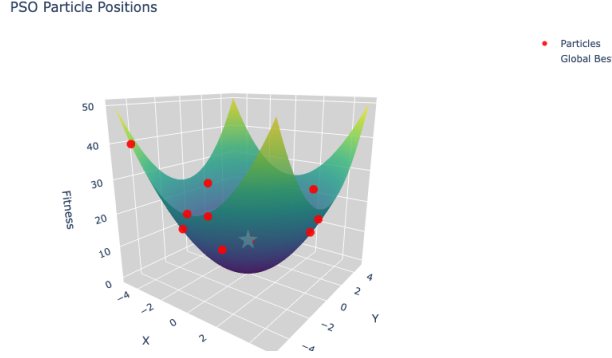


Figure 1: PSO'nun kavramsal görselleştirilmesi. Kırmızı noktalar, uygunluk yüzeyini keşfeden parçacıkları; beyaz yıldız ise global en iyi çözümü temsil etmektedir. Bu görselleştirme, parçacık hareketi ve global en iyi kavramının temel fikrini göstermektedir.

2.1.1 İşleyiş Mekanizması

PSO'nun işleyiş mekanizması, her parçacığın mevcut pozisyonunu, hızını ve şimdiye kadar ulaştığı en iyi pozisyonu korumasını içerir (Amini Fathian, 2020). Her iterasyonda, parçacığın hızı, keşif ve kullanımı dengeleyen stokastik bileşenleri içerecek şekilde güncellenir ve kişisel en iyi pozisyonu ile sürünün global en iyi pozisyonuna dayanır (Sarhani et al., 2022; Tomar et al., 2024). Bu hız güncelleme kuralı üç ana bileşeni bütünleştirir: parçacığın önceki hızı, parçacığı kendi en iyi bulduğu pozisyona geri yönlendiren bilişsel bileşen ve parçacığı sürünün global en iyi pozisyonuna çeken sosyal bileşen (Zambrano-Gutierrez et al., 2023). Hızın bu dinamik şekilde ayarlanması, parçacıkların arama alanının yeni bölgelerini keşfetmesini sağlarken, aynı zamanda sürü tarafından belirlenen umut verici bölgelere yakınsamasına olanak tanır (Munirah et al., 2020). Ardından, her parçacığın pozisyonu, bu değiştirilmiş hızın mevcut pozisyonuna eklenmesiyle güncellenir ve parçacık çok boyutlu arama alanında hareket eder (Dahal et al., 2024; Elshaboury, 2021). Bu yinelemeli süreç, önceden belirlenmiş bir durma kriteri karşılanana kadar devam eder; örneğin maksimum iterasyon sayısı veya tatmin edici bir çözüm kalitesi gibi (Hubálovský et al., 2023).

2.1.2 Uygulamalar

PSO, karmaşık çözüm alanlarında etkin bir şekilde gezinebildiği için mühendislik, finans ve makine öğrenimi gibi çeşitli alanlarda geniş uygulama bulmuştur; özellikle sürekli optimizasyon gerektiren problemlerde (Ajibade Ojeniyi, 2022). Örneğin, sinir ağı ağırlıklarının optimizasyonu, akıllı kontrol sistemlerinin tasarımı ve karmaşık görevlerin zamanlaması gibi alanlarda sıkça kullanılmaktadır (Bhakhar Chhillar, 2024; MUKUNZI et al., 2024). Ayrıca, dayanıklılığı ve uyarlanabilirliği, evrimsel hesaplama araştırmalarında özellikle global optimizasyon ve mühendislik problemleri için önemli bir odak noktası haline gelmesini sağlamaktadır (Cai et al., 2024). Algoritmanın popülasyon tabanlı yapısı, bireysel ve sosyal öğrenme yoluyla kolektif zekâyı taklit ederek, çok boyutlu arama alanlarında optimal parametre değerlerini verimli bir şekilde araştırmasını mümkün kılar (Cheng et al., 2018; Sohail et al., 2014). PSO, kaynak tahsisini optimize etmek için de etkin bir şekilde uygulanmıştır; buna üretim görevlerinin zamanlaması ve kablolu ağlarda enerji dağıtımının yönetimi gibi senaryolar dahildir (Kumar et al., 2023). Algoritmanın çok yönlülüğü, kimyasal sistemler, robotik ve görüntü işleme gibi çeşitli zorlu optimizasyon problemlerinde etkili kullanımı sağlar (Yuan Yin, 2015). Bunların ötesinde, PSO enerji sistemleri, havacılık mühendisliği ve yapay zekâ gibi alanlarda da yararlılığını göstermiş olup, çeşitli araştırma alanlarına geniş uygulanabilirliğini ortaya koymaktadır (Fang et al., 2023).

2.1.3 Avantajlar ve Dezavantajlar

PSO'nun başlıca avantajlarından biri, minimal parametre gereksinimleri ve türevsiz yapısı sayesinde hızlı yakınsama ve hesaplama verimliliği sağlamasıdır; bu durum hesaplama maliyetlerini azaltır (Liu et al., 2024). Bilişsel ve sosyal bileşenlerin etkileşimi yoluyla keşif ve kullanımı etkin bir şekilde dengeleme yeteneği, PSO'nun çeşitli alanlarda yaygın olarak benimsenmesine katkıda bulunur (Utkarsh Jain, 2024; Cai et al., 2024). Buna karşılık, PSO bazen yüksek boyutlu veya çok modlu çözüm alanlarında erken yakınsamadan etkilenebilir; keşif ve kullanım dengesinin dikkatli yönetilmemesi durumunda optimal olmayan çözümlere yol açabilir (Mogale et al., 2017). Bunu hafifletmek için, gelişmiş PSO varyantları, parçacıkların yerel optimumlara çok hızlı yakınsamalarını önlemek amacıyla hız güncelleme kuralını iyileştiren ataletsel ağırlıklar veya kısıtlama faktörleri gibi mekanizmalar sunar (Weiel et al., 2021; Hernández-Briones et al., 2024). Örneğin, ataletsel ağırlık, global ve yerel arama yeteneklerini dengelemeye hizmet eder ve genellikle sonraki iterasyonlarda yerel aramayı güçlendirmek için zamanla azalır (Naderi et al., 2022). Bu uyarlanabilir mekanizma, genellikle dikkatli parametre ayarlamalarıyla birlikte, algoritmanın performansı ve elde edilen çözümlerin güvenilirliği üzerinde önemli bir etkiye sahiptir (Innocente Sienz, 2006).

2.2 Differential Evolution(DE)

Diferansiyel Evrim Algoritması, benzersiz mutasyon ve çaprazlama işlemleri ile arama alanını verimli bir şekilde keşfetmek için vektör farklarını kullanan bir başka popülasyon tabanlı meta-sezgisel algoritmadır (Adediran Ameen, 2024). Storn ve Price tarafından 1997 yılında geliştirilen DE, aday çözümlerden oluşan bir popülasyon üzerinde çalışır ve bu çözümleri yeni ve geliştirilmiş çözümler üretmek için mutasyon, yeniden birleşim ve seçim süreçleriyle yinelemeli olarak iyileştirir (Munirah et al., 2020; Zhang et al., 2024). Özellikle sürekli optimizasyon problemleri için etkilidir ve kendi kendine uyarlanabilen yapısı ve basitleştirilmiş operatör yapısı sayesinde geleneksel genetik algoritmalara göre bir iyileştirme olarak kabul edilir (Wang et al., 2024). Particle Swarm Optimization'a benzer şekilde, Differential Evolution 1990'ların ortasında ortaya çıkmış ve hızlı bir şekilde güçlü bir popülasyon tabanlı optimizasyon yöntemi olarak öne çıkmıştır (Piotrowski et al., 2023). Başlangıçtaki sadeliği, kısa sürede geniş ilgi çekmiş ve son çeyrek yüzyılda çeşitli bilimsel ve mühendislik disiplinlerinde birçok varyant ve uygulamanın geliştirilmesini teşvik etmiştir (Munirah et al., 2020; Piotrowski et al., 2023).

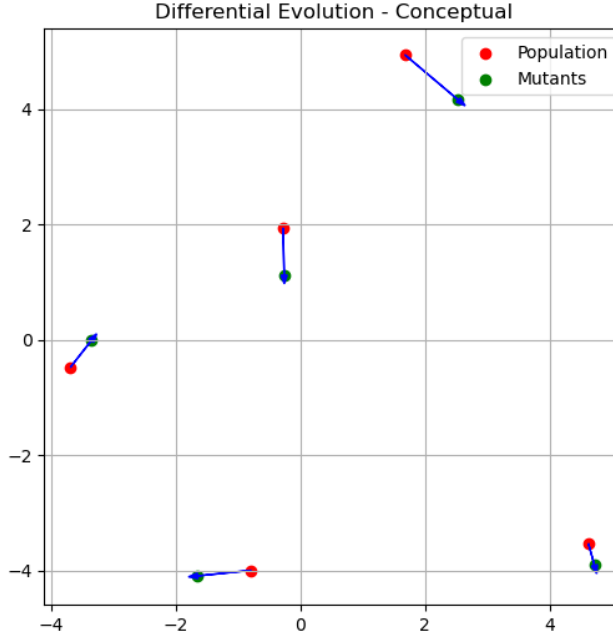


Figure 2: Differential Evolution (DE) algoritmasının kavramsal görselleştirilmesi. Kırmızı noktalar, mevcut aday çözüm popülasyonunu; yeşil noktalar, popülasyon üyeleri arasındaki farklardan oluşturulan mutant vektörleri; mavi oklar ise mutasyon yönlerini temsil etmektedir. Bu görsel, vektör farkları yoluyla yeni aday çözümlerin oluşturulmasının temel ilkesini göstermektedir.

2.2.1 İşleyiş Mekanizması

Differential Evolution algoritmasının işleyiş mekanizması, mevcut popülasyondaki her birey için, iki rastgele seçilen vektörün ölçeklendirilmiş farkı ile bir temel vektörün değiştirilmesi yoluyla bir mutant vektör oluşturulan döngüsel bir süreci içerir (Li et al., 2024). Bu yeni oluşturulan mutant vektör, hedef vektör ile bir çaprazlama işlemi geçirir ve bu işlem sonucunda bir deneme vektörü elde edilir (Schneider et al., 2019). Ardından, bir seçim süreci, deneme vektörünün bir sonraki nesilde hedef vektörün yerine geçip geçmeyeceğini, fitness değerlerine dayanarak belirler ve böylece popülasyonu daha iyi çözümlere yönlendirir. Farklılık temelli değişimlere vurgu yapan bu yinelemeli iyileştirme süreci, DE'nin karmaşık ve doğrusal olmayan arama alanlarında etkin bir şekilde gezinmesini ve optimal çözümlere yakınsamasını sağlar (Brest Maučec, 2025). Algoritmanın etkinliği, popülasyon boyutu, mutasyon faktörü ve çaprazlama oranı gibi hiperparametrelerin uygun seçimi ve ayarlanmasına kritik ölçüde bağlıdır; bu parametreler performans ve yakınsama özelliklerini derinden etkiler (Jiang et al., 2024; Zhang et al., 2024). Bu parametreler, keşif ve kullanım dengesini belirler ve algoritmanın çözüm alanını kapsamlı bir şekilde aramasını ve aynı zamanda optimal bölgelere hızlı yakınsamasını sağlar (Sharifi-Noghabi et al., 2016). Standart Differential Evolution algoritması, sağlam olmasına rağmen, üç temel kontrol parametresini içerir: popülasyon boyutu, mutasyon ölçekleme faktörü ve çaprazlama oranı; bu parametrelerin optimal değerleri probleminden probleme değişir (Khalfi et al., 2021). Bu parametrelerin ayarlanması, adaptif kontrol mekanizmaları aracılığıyla basitleştirilebilir ve algoritmanın arama süreci boyunca kendini dinamik olarak ayarlayarak daha hızlı ve güvenilir yakınsama sağlamasına olanak tanır (Charles Parks, 2018).

2.2.2 Uygulamalar

Differential Evolution, ortaya çıktığı günden bu yana çeşitli alanlarda son derece değerli olduğunu kanıtlamış ve mühendislik, veri bilimi ve bilimsel araştırmalardaki karmaşık optimizasyon problemlerini başarıyla çözmüştür (Azzam et al., 2024; Bilal et al., 2020). Geniş uygulanabilirliği, sinyal işleme, enerji sistemleri optimizasyonu ve finansal modelleme gibi alanlara kadar uzanmaktadır (Bilal et al., 2020). Ayrıca, DE'nin uyarlanabilirliği, gen ekspresyon analizi için biyoinformatik ve karmaşık anten sistemlerinin tasarımı gibi son derece özel alanlarda kullanılmasına olanak tanır (Kononova et al., 2021). Bu metodoloji, üç temel operatör—mutasyon, çaprazlama ve seçim—kullanarak, nesiller boyunca iteratif bir şekilde giderek geliştirilmiş çözümler üretir (Du et al., 2016).

2.2.3 Avantajlar ve Dezavantajlar

Differential Evolution (DE), arama metodolojisi açısından Particle Swarm Optimization (PSO) gibi algoritmalarından temel olarak ayrılır. PSO'da parçacıklar her nesilde yeni çözümleri değerlendirir ve yalnızca yeni çözüm daha iyiye pozisyonlarını günceller; oysa DE farklı bir strateji benimser: Bir birey her nesilde yeni bir çözümü değerlendirir ve yalnızca yeni çözüm daha kötü değilse

pozisyonunu günceller (Napiórkowski et al., 2022). Bu özellik, DE'nin yerel optimumlardan kaçınmasını önemli ölçüde kolaylaştırır ve karmaşık optimizasyon alanlarında yaygın olan başlangıç değerlerine duyarlılığı azaltır (Mousavirad et al., 2024). Differential Evolution'un önemli bir avantajı, güçlü global arama yetenekleri ve diğer evrimsel algoritmalara kıyasla daha az kontrol parametresi ile geniş bir optimizasyon problemi yelpazesinde sağlam performans göstermesidir (Farinati Vanneschi, 2024). DE'nin kendini düzenleyebilen yapısı ve doğrusal olmayan, konveks olmayan ve gürültülü hedef fonksiyonları yönetebilme kapasitesi, onu gerçek dünya optimizasyon problemlerini çözmek için esnek bir araç hâline getirir (Zhong et al., 2024). Bu popülasyon tabanlı teknik, aday çözümleri yinelemeli olarak iyileştirir ve verilen problem için optimal çözüme yakınsar; kendine özgü mutasyon, çaprazlama ve seçim mekanizmalarıyla diğer algoritmalarından ayrılır (Li Zhang, 2023). Bununla birlikte, DE bazen yakınsama sırasında istikrarsızlık yaşayabilir (Munirah et al., 2020) ve özellikle açgözlü uygulamalarda popülasyondaki çeşitliliği kaybetmeye meyilli olabilir (Charles Parks, 2018). DE'nin performansı ayrıca popülasyon boyutu, mutasyon ölçekleme faktörü ve çaprazlama oranı gibi kontrol parametrelerinin dikkatli ayarlanmasına büyük ölçüde bağlıdır; bu parametrelerin optimal değerleri problemde problemde değişebilir (Khalfi et al., 2021; Munirah et al., 2020).

2.3 Simulated Annealing (SA)

Simüle Tavlama Algoritması, metalurjideki tavlama sürecinden ilham alan bir meta-sezgisel optimizasyon algoritmasıdır; bu süreçte bir malzeme ısıtılır ve ardından kristal boyutunu artırmak ve kusurları azaltmak için yavaşça soğutulur (Das Suganthan, 2010). Optimizasyon problemlerine uygulandığında, bu benzetme algoritmanın özellikle aramanın başlangıç aşamalarında daha kötü çözümleri olasılıksal olarak kabul etmesine olanak tanır. Bu olasılıksal kabul, zamanla yavaşça azalan bir "sıcaklık" parametresi tarafından yönetilir ve SA'yı tamamen açgözlü yaklaşımlardan ayırır; bu sayede algoritma yerel optimumlardan kaçabilir ve çözüm alanını kapsamlı bir şekilde keşfedebilir (Gift Deza, 2021). Bu mekanizma, çözüm alanının daha ayrıntılı bir şekilde araştırılmasını kolaylaştırır ve sıcaklık azaldıkça ve alt-optimal hareketleri kabul etme olasılığı düştükçe, daha fazla sömürücü (exploitative) bir aramaya kademeli olarak geçiş sağlar.

2.3.1 İşleyiş Mekanizması

Simüle Tavlama algoritmasının temel mekanizması, arama alanındaki çözümler arasında geçiş yapmayı, her aşamada hedef fonksiyonu değerlendirmeyi ve yeni çözümler için kabul veya reddetme kararlarını hem çözümlerin kalitesine hem de mevcut "sıcaklığa" göre vermeyi içerir (Tang Wang, 2024). Başlangıçta, yüksek sıcaklıklarda algoritma alt-optimal çözümleri kabul etme eğilimi daha yüksektir; bu da arama alanının kapsamlı bir şekilde keşfedilmesini kolaylaştırır (Odeyemi Zhang, 2025). Sıcaklık, önceden belirlenmiş bir soğuma planına göre kademeli olarak azaldıkça, daha düşük kaliteli çözümleri kabul etme olasılığı

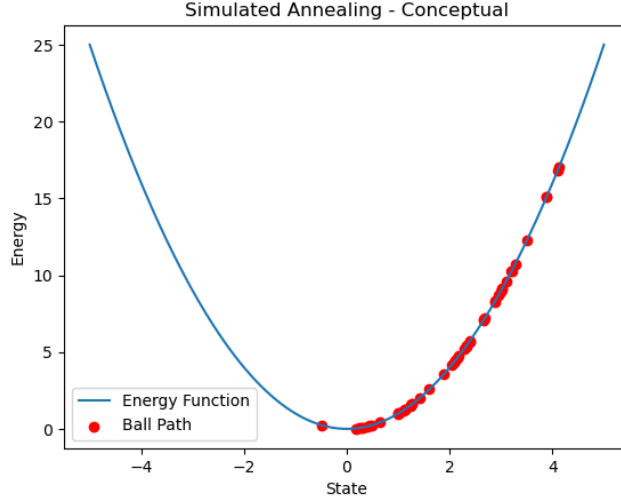


Figure 3: Simüle Tavlama (SA) algoritmasının kavramsal görselleştirilmesi. Kırmızı noktalar, enerji alanı üzerindeki ‘top’un zaman içindeki pozisyonlarını gösterirken, eğri uygunluk (fitness) veya enerji fonksiyonunu temsil eder. Bu şekil, SA’nın yüksek sıcaklıklarda olasılıksal olarak daha kötü çözümleri kabul edebildiğini ve zamanla ‘soğuyarak’ minimuma doğru yakınsadığını göstermektedir.

düşer; bu durum aramayı yerel optimumlara yönlendirir ve nihayetinde global optimuma yakınsamasını sağlar (İbrahim Younis, 2023; Pawłowski et al., 2022). Bu olasılık, hedef fonksiyondaki değişiklik ve mevcut sıcaklığa bağlı olarak alt-optimal bir çözümü kabul etme olasılığını nicelendirilen Boltzmann dağılımından türetilir (İbrahim Younis, 2023). Bu tavlama planı (annealing schedule) kritik bir bileşendir; çünkü sıcaklığın düşürülme hızını belirler ve algoritmanın yakınsama özelliklerini ile elde edilen nihai çözümün kalitesini önemli ölçüde etkiler (Suman Kumar, 2005). Ardışık iterasyonlarda soğuma planı aracılığıyla sıcaklığın sistematik olarak azaltılması, arama alanının keşfi üzerinde kontrol sağlar (Yan et al., 2023).

2.3.2 Uygulamalar

Simüle Tavlama algoritmasının olasılıksal özelliği, alt-optimal çözümleri kabul etme yeteneği, yerel minimumlardan kaçınmada ve karmaşık çözüm alanlarının kapsamlı bir şekilde keşfedilmesinde kritik bir rol oynar (Hunagund et al., 2017). SA, chiplet tabanlı yapay zekâ hızlandırıcılarının tasarımı ile grafik ve dizi optimizasyonu gibi çeşitli alanlarda uygulanmaktadır (Liu et al., 2021; Mishty Sadi, 2024). Ayrıca, SA farklı disiplinlerde karmaşık kombinatoriyal optimizasyon problemlerinin çözümünde etkinliğini kanıtlamıştır; bu disiplinler arasında ekonomi ve hesaplamalı bilimlerde bulunmaktadır (Naser Naser, 2024). Yüksek

boyutlu veri setlerinde genomik özellik seçimi süreçlerinde ilgili genleri belirlemede özellikle başarılıdır (Sinha et al., 2024). Geniş uygulama alanı, hedef fonksiyonların türevlenebilir olmadığı, süreksiz veya yüksek derecede çok modlu olduğu senaryoları da kapsar; bu tür durumlarda gradyan tabanlı yöntemlerin etkinliği sınırlıdır (Ibrahim Younis, 2023).

2.3.3 Avantajlar ve Dezavantajlar

Simüle Tavlama algoritmasının başlıca avantajlarından biri, teorik olarak sonsuz uzunlukta bir soğuma planıyla global optimumu bulma garantisidir; ancak pratik uygulamalarda hesaplama açısından uygulanabilirlik için sınırlı süreli planlar kullanılır (Liu et al., 2021). Bu teorik güce rağmen, algoritmanın parametre ayarlarına, özellikle soğuma planı ve başlangıç sıcaklığına duyarlılığı önemli bir pratik zorluk oluşturur; bu parametreler doğrudan yakınsama hızı ve çözüm kalitesini etkiler (Mishra, 2024; Vert et al., 2024). Bu parametre duyarlılığı, kapsamlı keşif için gerekli olan yavaş soğuma oranlarının hesaplama maliyeti ile birleştiğinde, gerçek dünya uygulamalarında pratik bir sınırlama oluşturur. Bu durum, keşif ve sömürü (exploitation) arasındaki dengeyi sağlamak açısından uygun bir soğuma planının seçimini kritik hâle getirir (Ball et al., 2018; Benfer et al., 2023). SA'nın alt-optimal hareketleri olasılıksal olarak kabul ederek yerel optimumlardan kaçabilme kapasitesi, özellikle karmaşık ve çok modlu arama alanlarında önemli bir özelliktir (Valadez-Vergara Szabó, 2024). Bununla birlikte, bu güçlü yön bazı potansiyel dezavantajlarla dengelenir; örneğin yüksek sistem yükü ve uzun göç süreleri (YANG, 2023). SA iyi çözümler bulabilse de, sınırlı bir sürede optimal çözüm garanti etmez ve performansı başlangıç çözümüne ve tavlama planının özel uygulanmasına büyük ölçüde bağlıdır (Kunzmann et al., 2020; Liu et al., 2021).

2.4 Tabu Search (TS)

Tabu Arama, kısa süreli bir bellek (tabu listesi) kullanarak yerel arama tekniklerini geliştirmeyi amaçlayan bir meta-sezgisel optimizasyon algoritmasıdır; bu bellek, yakın zamanda ziyaret edilen çözümlerin yeniden keşfedilmesini engeller (Abd et al., 2014). Bu sofistike yaklaşım, arama sürecini alt-optimal yerel çözümlerden uzaklaştırarak, tüm çözüm alanının daha geniş bir şekilde araştırılmasını teşvik eder (Rostami et al., 2021). Bu, algoritmayı daha önce karşılaşılan çözümlere geri götürecek belirli hareketleri sistematik olarak yasaklayarak sağlar; bu da çeşitlendirmeyi teşvik eder ve tekrarlayan arama yollarından kaçınmayı sağlar (Chen et al., 2024; Amini Fathian, 2020). Sonuç olarak, önceki çözümlerin yeniden ziyaret edilmesini önleyerek, Tabu Arama basit yerel arama algoritmalarının sıkça karşılaştığı ve sıklıkla yerel optimumlarda sıkışma problemine etkili bir çözüm sunar (Okechukwu et al., 2024). Bu kasıtlı kısıtlama, algoritmanın çözüm alanında daha verimli bir şekilde gezinmesini ve yeni bölgeler keşfetmesini mümkün kılar (Caballero-Martin et al., 2024).

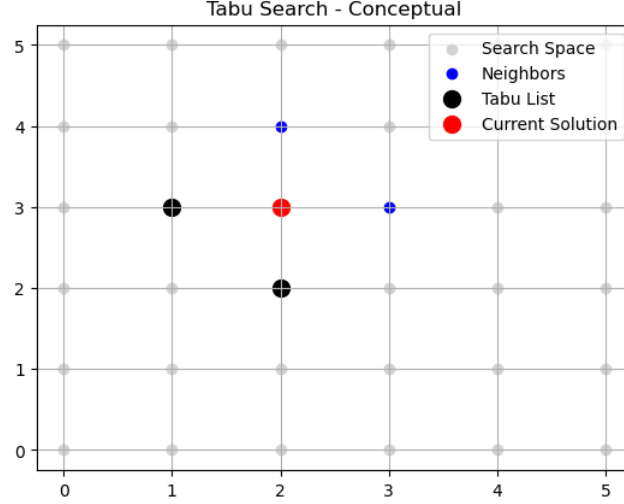


Figure 4: Tabu Arama'nın kavramsal görselleştirilmesi. Kırmızı nokta mevcut çözümü, mavi noktalar komşu aday çözümleri ve siyah noktalar tabu listesinde yasaklanmış çözümleri temsil eder. Bu görselleştirme, yakın çevrenin araştırılması ve yakın zamanda ziyaret edilen çözümlerden kaçınılması temel fikrini aktarır.

2.4.1 İşleyiş Mekanizması

Tabu Search, mevcut bir çözümden geliştirilmiş bir komşu çözüme iteratif olarak geçiş yaparak çalışır ve bu süreçte merkezi olarak “tabu listesi” kullanarak yakın zamanda ziyaret edilen çözümleri veya hareketleri kaydeder (YANG, 2023). Bu liste, kısa süreli bir bellek işlevi görerek algoritmanın döngüye girmesini önler ve arama alanında yeni bölgelerin keşfedilmesini teşvik eder (YANG, 2023). Tabu listesindeki çözümlere yol açacak hareketler yasaklanmıştır; ancak, eğer bir arz kriteri (aspiration criterion) karşılanırsa, daha önce keşfedilen herhangi bir çözümden üstün bir çözüm elde ediliyorsa tabu hareketine izin verilir (Augusto et al., 2021). Bu iteratif süreç, maksimum iterasyon sayısına ulaşılması, önceden belirlenmiş bir hesaplama bütçesinin tükenmesi veya uzun süreli iyileşme eksikliği gibi bir sonlandırma koşulu gerçekleşene kadar devam eder (Amini Fathian, 2020).

2.4.2 Uygulamalar

Tabu Search, çeşitli zamanlama problemleri, araç yönlendirme ve özellikle yapısal optimizasyon da dahil olmak üzere geniş bir kombinatoryal optimizasyon problemleri yelpazesinde başarıyla uygulanmıştır (Ghaemifard Ghannadiasl, 2024). Farklı senaryolara uyum sağlama yeteneği, onu karmaşık problemleri çözmede güçlü bir araç haline getirir (Li Ai, 2023). Örneğin, proje süresini minimize etmek için kaynak kısıtlı proje zamanlama problemlerinde etkili bir şekilde kul-

lanılmıştır (Golab et al., 2021) ve zaman pencereleri ile çoklu depo dönüşlü petrol kamyonu yönlendirmesini optimize etmekte uygulanmıştır (Wang et al., 2019). Ayrıca, Tabu Arama, ağ tasarımı, telekomünikasyon ve biyoinformatik gibi verimli tahsis ve sıralama gerektiren alanlarda da uygulanmakta, çok yönlülüğünü göstermektedir (Molina et al., 2020). Lagrange tabanlı yöntemler gibi diğer sezgisel yaklaşımlarla entegre edilebilme yeteneği, karmaşık bağımlılıkları ve kısıtları ele almada etkinliğini daha da artırmaktadır (Niroumandrad et al., 2024).

2.4.3 Avantajlar ve Dezavantajlar

Tabu Search'ün önemli bir avantajı, yakın zamanda ziyaret edilen çözümlerin bir hafızasını kullanarak yerel optimumlardan kaçabilme kapasitesidir; bu hafıza kısa, orta ve uzun vadeli yapıları içerebilir ve çözüm alanının daha kapsamlı bir şekilde araştırılmasını sağlayarak alt-optimal bölgelerin tekrarlayan şekilde keşfini önler (Pelleau et al., 2009; Yang Burn, 2019). Bununla birlikte, temel bir zorluk, uygun tabu süre değerlerinin dikkatli seçilmesidir; çünkü çok kısa bir süre döngüsel aramalara yol açabilirken, aşırı uzun bir süre aramayı gereksiz şekilde kısıtlayabilir ve yakınsamayı yavaşlatabilir (Niroumandrad et al., 2024; Shahmanzari Aksen, 2020). Tabu Search'ün etkinliği ayrıca komşuluk yapısının ve arzu kriterlerinin tanımına da büyük ölçüde bağlıdır; bu yapılar sırasıyla izin verilen hareketleri ve tabu kurallarına istisnaları belirler (Díaz et al., 2014; Shahmanzari Aksen, 2020). Bazı uygulamalarda, tabu listesinin boyutu dinamik olabilir, alt ve üst sınırlar arasında değişebilir ve önemli ölçüde daha iyi bir çözüm bulunduğunda temizlenebilir; bu, aramanın yeniden başlatılmasına benzer (Gómez et al., 2023). Ayrıca, bir hareketin tabu durumu, daha önce ziyaret edilen çözümlerden daha yüksek bir hedef fonksiyon değerine yol açıyorsa göz ardı edilebilir (Pacheco et al., 2023). Tabu Search, özellikle kare atama problemi gibi hesaplama açısından zorlayıcı problemler için yüksek kaliteli çözümler bulmada başarılı olsa da, çok sayıda yerel optimumun bulunduğu senaryolarda yerel optimumlardan verimli kaçış sağlanamaması performansını olumsuz etkileyebilir ve bu durum yavaş yakınsamaya veya alt-optimal çözümlere yol açabilir (Abel Siraj, 2024; Ghnatios et al., 2019).

3 Algoritmaların Karşılaştırmalı Analizi

Her bir meta-sezgisel algoritma kendine özgü güçlü ve zayıf yönlere sahiptir; dolayısıyla etkinlikleri problem türüne, boyutuna ve hesaplama gereksinimlerine bağlıdır.

Table 1: Algoritmaların Performans Ölçütleri ve Karakteristikleri

Ölçüt / Kriter	PSO	DE	SA	TS
Arama Türü	Popülasyon tabanlı	Popülasyon tabanlı	Tek çözüm tabanlı	Tek çözüm tabanlı
Yakınsama Hızı	Hızlı (erken yakınsama riski)	Orta-hızlı (denge sağlanmış)	Yavaş (soğuma oranına bağlı)	Orta (iteratif ilerleme)
Küresel Optimuma Ulaşma	Orta-iyi, erken yakınsamaya eğilimli	Yüksek — güçlü küresel optimum potansiyeli	Yüksek (yeterli iterasyon ile)	Orta — tabu listesi yerel optimumlardan kaçmayı destekler
Keşif Yeteneği	Başlangıçta yüksek, sonra azalır	Yüksek — mutasyon çeşitliliği korur	Orta — yüksek sıcaklıklarda iyidir	Orta — tabu mekanizması ile çeşitlilik korunur
Sömürü (Intensification)	Orta	Güçlü — bireylerin kombinasyonu sonuçları iyileştirir	Güçlü — düşük sıcaklıkta etkili yerel arama	Çok güçlü — derin komşuluk araması
Hesaplama Maliyeti	Düşük-orta	Orta	Düşük	Orta
Parametre Hassasiyeti	Orta	Orta-yüksek	Yüksek	Düşük-orta
Yakınsama Kararlılığı	Orta — parametrelere bağlı	Yüksek	Orta	Orta
Kesikli / Sürekli Problemlere Uygunluk	Sürekli problemler	Sürekli optimizasyon	Her iki türde	En uygun kesikli
Parallellendirilebilirlik	Yüksek	Yüksek	Düşük	Düşük-orta
Erken Yakınsama Riski	Yüksek	Düşük	Düşük	Düşük-orta

Table 2: Algoritmaların Farklı Problem Türlerine Uygunluğu

Problem Türü / Alan	PSO	DE	SA	TS
Sürekli Optimizasyon	Çok uygun	Mükemmel	Uygun	Sınırlı uygunluk
Kesikli / Kombinatoriyal	Sınırlı — adaptasyon gerekli	Sınırlı — ikili/tam sayı versiyonu	Uygun	Mükemmel — planlama, yönlendirme, sıralama
Çok modlu Problemler	İyi, erken yakınsamaya eğilimli	Mükemmel — çeşitlilik güçlü	Etkili — stokastik kaçış	Orta — çeşitlilik kontrolü sınırlı
Çok Amaçlı Optimizasyon	Kolayca genişletilebilir	Güçlü — MODE iyi performans	Ek stratejiler gerekli	Genellikle tek amaç odaklı
Dinamik / Zamanla Değişen	Popülasyon güncellemeleri ile uyumlu	Yeniden başlatma gerekebilir	Sıcaklık değişimi ile uyum	İdeal değil — statik yapı
Kısıtlı Optimizasyon	Ceza fonksiyonları kullanılabilir	Esnek	Basit kısıtlarla	Tabu hafızası ile güçlü
Deterministik	Güvenilir	Yüksek doğruluk	Rastgelelik nedeniyle değişken	Stabil
Stokastik	Rastgeleliği iyi yönetir	Mutasyon çeşitliliği ile etkili	Uygun	Daha az esnek
Gerçek Zaman / Hız	Mükemmel	Dengeli	Yavaş	Orta

Table 3: Hesaplama Karmaşıklığı ve Ölçeklenebilirlik

Ölçüt / Kriter	PSO	DE	SA	TS
Temel Hesaplama Karmaşıklığı	$O(N_p \times D \times I)$	$O(N_p \times D \times I)$	$O(D \times I)$	$O(N_{neigh} \times I)$
Bellek Kullanımı	Düşük-orta	Orta	Çok düşük	Orta
İterasyon Başına Zaman	Düşük — paralel hesaplanabilir	Orta — ek maliyet	Düşük	Orta — komşuluk ve tabu kontrolleri
Ölçeklenebilirlik	İyi — paralelleştirme avantajı	İyi — büyük popülasyonlarda yavaşlayabilir	Orta — problem büyüdükçe yakınsama süresi artar	Sınırlı — büyük komşuluk maliyeti
Parallellendirilebilirlik	Çok yüksek	Çok yüksek	Düşük	Düşük-orta
Yüksek Boyutlu Problemler	Orta — erken yakınsama riski	Yüksek — stabil	Orta-düşük — keşif yavaş	Düşük — büyük komşuluk maliyeti
Hesaplama Verimliliği	Yüksek	Orta-yüksek	Yüksek	Orta
Gürültülü / Karmaşık Fonksiyonlar	Orta — çeşitlilik kaybı	Yüksek	Orta — rastgelelik tutarsız	Yüksek — deterministik yapı

4 Sonuç

Uygun meta-sezgisel algoritmanın seçimi, her yöntemin kendine özgü güçlü yönleri ve ödünleşimleri bulunduğundan, problem türü, boyutluluk (dimensionality) ve hesaplama kısıtlarının dikkatli bir şekilde değerlendirilmesini gerektirir. Bu karşılaştırmalı analiz, Parçacık Sürü Optimizasyonu (PSO), Diferansiyel Evrim (DE), Benzetilmiş Tavlama (SA) ve Tabu Arama (TS) algoritmalarının çeşitli optimizasyon problem özellikleri üzerindeki güçlü ve zayıf yönlerini ayrıntılı biçimde ortaya koymaktadır. Yapılan değerlendirme, hiçbir meta-sezgisel algoritmanın tüm problem alanlarında evrensel olarak diğerlerinden üstün performans sergilemediğini göstermektedir. Popülasyon tabanlı yöntemler olan PSO ve DE, özellikle sürekli (continuous) ve yüksek boyutlu optimizasyon problemlerinde üstün ölçeklenebilirlik ve paralelleştirme yetenekleri sergiler. Bunlar arasında DE, keşif (exploration) ve sömürü (exploitation) arasında daha dengeli bir ilişki kurarak, PSO'da sıkça gözlemlenen erken yakınsama (premature convergence) riskini etkili biçimde azaltır. SA, yakınsama açısından daha yavaş olsa da, stokastik (olasılıksal) ve çok modlu (multi-modal) problemlerde sağlam bir performans sunar; olasılıksal kabul mekanizması sayesinde yerel optimumlardan etkili biçimde kaçabilir. Buna karşın TS, ayrık (discrete) ve kombinatoriyal optimizasyonda öne çıkar; hafıza tabanlı arama yapısı sayesinde daha önce keşfedilen bölgeleri yeniden ziyaret etmekten kaçınır. Ancak, ardışık (sequential) ve deterministik doğası hem ölçeklenebilirliği hem de dinamik ortamlara uyarlatabilirliği sınırlayabilir. Genel olarak, DE sürekli optimizasyon problemleri için son derece çok yönlü bir algoritma olarak öne çıkarken, TS kombinatoriyal ortamlarda üstünlük göstermektedir. SA rastgeleliğe karşı dayanıklılık sunarken, PSO hızlı yakınsama veya hesaplama verimliliği gerektiren senaryolarda güçlü bir seçenek olmaya devam etmektedir.

4.1 Öneriler

Bu analiz temelinde, bir meta-sezgisel algoritma seçimi problem türü ve hesaplama gereksinimlerine göre yönlendirilmelidir. DE, küresel arama kabiliyeti sayesinde sürekli ve yüksek boyutlu problemler için önerilmektedir. PSO, hızlı yakınsama veya paralel hesaplama avantajı istendiğinde uygundur; ancak erken yakınsama riskine karşı dikkatli olunmalıdır. SA, yerel minimumlardan güvenilir biçimde kaçabilme özelliğiyle stokastik veya çok modlu problemlerde idealdir; ancak daha yavaş yakınsar. TS, özellikle zamanlama (scheduling), rotalama (routing) veya sıralama (sequencing) gibi ayrık ve kombinatoriyal problemler için oldukça etkilidir; hafıza tabanlı arama ve stratejik kaçınma mekanizmalarıyla önceki çözümlerin tekrarlanması engeller. Sonuç olarak, uygun algoritmanın seçimi; yakınsama hızı, çözüm kalitesi, hesaplama kaynakları ve probleme özgü özellikler arasındaki dengeye bağlıdır.

5 Üretken Yapay Zeka Kullanımı

Bu çalışmada üretken yapay zekâ araçları yalnızca dilbilgisi düzeltme, LaTeX şablon oluşturma ve çeviri amacıyla kullanılmıştır. GitHub Copilot, Markdown dosyalarının hazırlanması ve Plotly entegrasyonu süreçlerinde yardımcı olmuştur. Bu raporda sunulan analitik içgörüler ve içerik, optimizasyon algoritmalarına ilişkin akademik literatürün kapsamlı bir incelemesi ve sentezi temelinde yazarın özgün çalışmasıdır.

References

- [1] Abd, G., Mahmoud, A. M., & El-Sayed, M. H. (2014). A Comparative Study of Meta-heuristic Algorithms for Solving Quadratic Assignment Problem. *International Journal of Advanced Computer Science and Applications*, 5(1). <https://doi.org/10.14569/ijacsa.2014.050101>
- [2] Abel, E. W., & Siraj, S. (2024). Investigating fairness in decisions involving ordinal classification: A COVID-19 case study. *Research Portal Denmark*. <https://local.forskningportal.dk/local/dki-cgi/ws/cris-link?src=sduid=sdu-5b3b1ec0-9758-4bc4-88d8-9eafa6dc1ab9>
- [3] Adediran, E. M., & Ameen, S. (2024). Machine Learning and Optimization Techniques for Solving Inverse Kinematics in a 7-DOF Robotic Arm. *arXiv preprint*. <https://doi.org/10.48550/arxiv.2406.13064>
- [4] Ajibade, S.-S. M., & Ojeniyi, A. (2022). Bibliometric Survey on Particle Swarm Optimization Algorithms (2001–2021). *Journal of Electrical and Computer Engineering*, 2022, 1–12. <https://doi.org/10.1155/2022/3242949>
- [5] Amini, M. M., & Fathian, M. (2020). Optimizing bid search in large outcome spaces for automated multi-issue negotiations using meta-heuristic methods. *Decision Science Letters*, 1–20. <https://doi.org/10.5267/j.dsl.2020.10.007>
- [6] Augusto, A., Dumas, M., Rosa, M. L., Leemans, S. J. J., & Broucke, S. K. L. M. vanden. (2021). Optimization framework for DFG-based automated process discovery approaches. *Software & Systems Modeling*, 20(4), 1245–1270. <https://doi.org/10.1007/s10270-020-00846-x>
- [7] Azzam, S. M., Emam, O. E., & Abolaban, A. S. (2024). An improved Differential evolution with Sailfish optimizer (DESFO) for handling feature selection problem. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-63328-w>
- [8] Balkaya, Ç., Göktürkler, G., Ekinci, Y. L., & Karaoğlu, S. T. (2014, March). Metaheuristics in applied geophysics. In *Proceedings of the 15th European Metaheuristics Community Workshop: Metaheuristics and Engineering*, Istanbul. <https://doi.org/10.13140/2.1.4454.7526>

- [9] Ball, R. C., Branke, J., & Meisel, S. (2018). Optimal Sampling for Simulated Annealing Under Noise. *INFORMS Journal on Computing*, 30(1), 200–215. <https://doi.org/10.1287/ijoc.2017.0774>
- [10] Benfer, M., Heyer, V., Brützel, O., Liebrecht, C., Peukert, S., & Lanza, G. (2023). Analysis of metaheuristic optimisation techniques for simulated matrix production systems. *Production Engineering*, 18(1), 159–168. <https://doi.org/10.1007/s11740-023-01225-3>
- [11] Bhakhar, R., & Chhillar, R. S. (2024). Dynamic multi-criteria scheduling algorithm for smart home tasks in fog-cloud IoT systems. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-81055-0>
- [12] Bilal, Pant, M., Zaheer, H., García-Hernández, L., & Abraham, A. (2020). Differential Evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence*, 90, 103479. <https://doi.org/10.1016/j.engappai.2020.103479>
- [13] Bolufé-Röhler, A., & Tamayo-Vera, D. (2025). Machine Learning for Enhancing Metaheuristics in Global Optimization: A Comprehensive Review. <https://doi.org/10.20944/preprints202507.2611.v1>
- [14] Brest, J., & Maučec, M. S. (2025). Comparative Study of Modern Differential Evolution Algorithms: Perspectives on Mechanisms and Performance. <https://doi.org/10.20944/preprints202503.1551.v1>
- [15] Caballero-Martin, D., López-Guede, J. M., Estévez, J., & Graña, M. (2024). Artificial Intelligence Applied to Drone Control: A State of the Art. *Drones*, 8(7), 296. <https://doi.org/10.3390/drones8070296>
- [16] Cai, T., Zhang, S., Ye, Z., Wen, Z., Wang, M., He, Q., Chen, Z., & Bai, W. (2024). Cooperative metaheuristic algorithm for global optimization and engineering problems inspired by heterosis theory. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-78761-0>
- [17] Charles, A., & Parks, G. T. (2018). Application of Differential Evolution algorithms to multi-objective optimization problems in mixed-oxide fuel assembly design. *Annals of Nuclear Energy*, 127, 165–177. <https://doi.org/10.1016/j.anucene.2018.12.002>
- [18] Chen, B., Cao, L., Chen, C., Chen, Y., & Yue, Y. (2024). A comprehensive survey on the chicken swarm optimization algorithm and its applications: state-of-the-art and research challenges. *Artificial Intelligence Review*, 57(7). <https://doi.org/10.1007/s10462-024-10786-3>
- [19] Cheng, S., Liu, B., Ting, T. O., Huang, K., et al. (2016, July). Survey on data science with population-based algorithms. *Big Data Analytics*, 1(1), 1–20. <https://doi.org/10.1186/s41044-016-0003-3>

- [20] Cheng, S., Lu, H., Lei, X., & Shi, Y. (2018). A quarter century of particle swarm optimization. *Complex & Intelligent Systems*, 4(3), 227–239. <https://doi.org/10.1007/s40747-018-0071-2>
- [21] Cuevas, E., Ascencio-Piña, C., Pérez, M., & Morales-Castañeda, B. (2024). Considering radial basis function neural network for effective solution generation in metaheuristic algorithms. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-67778-0>
- [22] Dahal, S., Heggliid, G. J., Nøland, J. K., Chhetri, B. B., & Øyvang, T. (2024). Integrating Multiple Slack Bus Operations and Metaheuristic Techniques for Power Flow Optimization. *Research Square*. <https://doi.org/10.21203/rs.3.rs-5360850/v1>
- [23] Das, S., & Suganthan, P. N. (2010). Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*, 15(1), 4–31. <https://doi.org/10.1109/tevc.2010.2059031>
- [24] Díaz, I. M., Nasini, G., & Severín, D. (2014). A Tabu Search Heuristic for the Equitable Coloring Problem. In *Lecture notes in computer science* (p. 347–358). Springer. https://doi.org/10.1007/978-3-319-09174-7_30
- [25] Du, W., Leung, S. Y. S., Tang, Y., & Vasilakos, A. V. (2016). Differential Evolution With Event-Triggered Impulsive Control. *IEEE Transactions on Cybernetics*, 47(1), 244–257. <https://doi.org/10.1109/tcyb.2015.2512942>
- [26] Elshaboury, N. (2021). Investigating the occupant existence to reduce energy consumption by using a hybrid artificial neural network with metaheuristic algorithms. *Decision Science Letters*, 11(1), 91–104. <https://doi.org/10.5267/j.dsl.2021.8.001>
- [27] Fang, J., Liu, W., Chen, L.-W., Lauria, S., Miron, A., & Liu, X.-H. (2023). A Survey of Algorithms, Applications and Trends for Particle Swarm Optimization. *International Journal of Network Dynamics and Intelligence*, 24–50. <https://doi.org/10.53941/ijndi0201002>
- [28] Farinati, D., & Vanneschi, L. (2024). A survey on dynamic populations in bio-inspired algorithms. *Genetic Programming and Evolvable Machines*, 25(2). <https://doi.org/10.1007/s10710-024-09492-4>
- [29] Ghaemifard, S., & Ghannadiasl, A. (2024). A Comparison of Metaheuristic Algorithms for Structural Optimization: Performance and Efficiency Analysis. *Advances in Civil Engineering*, 2024(1). <https://doi.org/10.1155/2024/2054173>
- [30] Ghnatios, C., Hage, R.-M., & Hage, I. S. (2019). An efficient Tabu-search optimized regression for data-driven modeling. *Comptes Rendus Mécanique*, 347(11), 806–816. <https://doi.org/10.1016/j.crme.2019.11.006>

- [31] Gift, N., & Deza, A. (2021). *Practical MLOps Operationalizing Machine Learning Models*. O'Reilly Media.
- [32] Golab, A., Gooya, E. S., Falou, A. A., & Cabon, M. (2021). Review of conventional metaheuristic techniques for resource-constrained project scheduling problem. *Journal of Project Management*, 7(2), 95–110. <https://doi.org/10.5267/j.jpm.2021.10.002>
- [33] Gómez, P. G., Vela, C. R., & González-Rodríguez, I. (2023). Neighbourhood search for energy minimisation in flexible job shops under fuzziness. *Natural Computing*, 22(4), 685–704. <https://doi.org/10.1007/s11047-023-09967-w>
- [34] Hernández-Briones, M. A., Reséndiz-Flores, E. O., & Serrano-Toledo, J. L. (2024). A novel intelligent prediction approach with optimal dimensional reduction on shrinkage defects in iron casting. *Research Square*. <https://doi.org/10.21203/rs.3.rs-5182557/v1>
- [35] Hubálovský, Š., Hubálovská, M., & Matoušová, I. (2023). A New Hybrid Particle Swarm Optimization–Teaching–Learning–Based Optimization for Solving Optimization Problems. *Biomimetics*, 9(1), 8. <https://doi.org/10.3390/biomimetics9010008>
- [36] Hunagund, I. B., Pillai, V. M., & Kempaiah, U. N. (2017). A simulated annealing algorithm for unequal area dynamic facility layout problems with flexible bay structure. *International Journal of Industrial Engineering Computations*, 307–330. <https://doi.org/10.5267/j.ijiec.2017.8.004>
- [37] Ibrahim, O. A. S., & Younis, E. M. G. (2023). A Hybrid-Box tool for Explainable and Interactive Simulated Annealing Strategy. *Research Square*. <https://doi.org/10.21203/rs.3.rs-3578275/v1>
- [38] Innocente, M. S., & Sienz, J. (2006). Particle Swarm Optimization: Development of a General-Purpose Optimizer. *arXiv preprint*, 203–223. <https://doi.org/10.48550/arXiv.2101.09835>
- [39] Jiang, R., Du, G., Yu, S., Guo, Y., Goh, S. K., & Tang, H.-K. (2024). CADE: Cosine Annealing Differential Evolution for Spiking Neural Network. *arXiv preprint*. <https://doi.org/10.48550/arxiv.2406.02349>
- [40] Khalfi, S., Draa, A., & Iacca, G. (2021). A compact compound sinusoidal differential evolution algorithm for solving optimisation problems in memory-constrained environments. *Expert Systems with Applications*, 186, 115705. <https://doi.org/10.1016/j.eswa.2021.115705>
- [41] Kononova, A. V., Caraffini, F., & Bäck, T. (2021). Differential evolution outside the box. *Information Sciences*, 581, 587–604. <https://doi.org/10.1016/j.ins.2021.09.058>

- [42] Kumar, P., Kumar, K., Bohre, A. K., Adhikary, N., & Tesfaye, E. L. (2023). Intelligent planning of controllers for improved resilience in multi-area system involving nuclear power. *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-023-42155-5>
- [43] Kunzmann, P., Mayer, B. E., & Hamacher, K. (2020). Substitution matrix based color schemes for sequence alignment visualization. *BMC Bioinformatics*, 21(1). <https://doi.org/10.1186/s12859-020-3526-6>
- [44] Li, J., Majeed, A. P. P. A., & Lefevre, P. (2024). Halfway Escape Optimization: A Quantum-Inspired Solution for General Optimization Problems. *Research Square*. <https://doi.org/10.21203/rs.3.rs-5285753/v1>
- [45] Li, M., & Ai, J. (2023). Energy-Aware Clustering in the Internet of Things using Tabu Search and Ant Colony Optimization Algorithms. *International Journal of Advanced Computer Science and Applications*, 14(12). <https://doi.org/10.14569/ijacsa.2023.0141238>
- [46] Li, M., & Zhang, Y. (2023). Integrating Social Media Data and Historical Stock Prices for Predictive Analysis: A Reinforcement Learning Approach. *International Journal of Advanced Computer Science and Applications*, 14(12). <https://doi.org/10.14569/ijacsa.2023.0141203>
- [47] Liu, R., Yang, Z., Su, J., & Yu, C. (2024). A Hybrid Framework for Evaluating Financial Market Price: An Analysis of the Hang Seng Index Case Study. *International Journal of Advanced Computer Science and Applications*, 15(6). <https://doi.org/10.14569/ijacsa.2024.01506111>
- [48] Liu, X., Li, P., Meng, F., Zhou, H., Zhong, H., Zhou, J., Mou, L., & Song, S. (2021). Simulated annealing for optimization of graphs and sequences. *Neurocomputing*, 465, 310–324. <https://doi.org/10.1016/j.neucom.2021.09.003>
- [49] Martinez, A. D., Ser, J. D., Villar-Rodríguez, E., Osaba, E., Poyatos, J., Tabik, S., Molina, D., & Herrera, F. (2020). Lights and shadows in Evolutionary Deep Learning: Taxonomy, critical methodological analysis, cases of study, learned lessons, recommendations and challenges. *Information Fusion*, 67, 161–194. <https://doi.org/10.1016/j.inffus.2020.10.014>
- [50] Mishra, A. (2024). Machine Learning-Driven Optimization of TPMS Architected Materials Using Simulated Annealing. *arXiv preprint*. <https://doi.org/10.48550/arxiv.2406.05142>
- [51] Mishty, K., & Sadi, M. (2024). Chiplet-Gym: Optimizing Chiplet-based AI Accelerator Design with Reinforcement Learning. *arXiv preprint*. <https://doi.org/10.48550/arxiv.2406.00858>
- [52] Mogale, D. G., Kumar, S. K., & Tiwari, M. K. (2017). An MINLP model to support the movement and storage decisions of the Indian food grain supply chain. *Control Engineering Practice*, 70, 98–113. <https://doi.org/10.1016/j.conengprac.2017.09.017>

- [53] Molina, C. A., Salmerón, J. L., Eguía, I., & Racero, J. (2020). The heterogeneous vehicle routing problem with time windows and a limited number of resources. *Engineering Applications of Artificial Intelligence*, 94, 103745. <https://doi.org/10.1016/j.engappai.2020.103745>
- [54] Mousavirad, S. J., Schaefer, G., Rezaee, K., Oliva, D., Zabihzadeh, D., Chakraborty, R. K., Mohammadigheymasi, H., & Pedram, M. (2024). A novel metaheuristic population algorithm for optimising the connection weights of neural networks. *Evolving Systems*, 16(1). <https://doi.org/10.1007/s12530-024-09641-1>
- [55] MUKUNZI, W. K., Muganda, B. W., & Kasamani, B. S. (2024). A Machine Learning Model for Algorithmic Optimization of Superannuation Schemes. *Research Square*. <https://doi.org/10.21203/rs.3.rs-5280421/v1>
- [56] Munirah, N. A., Remli, M. A., Mohd, N., Wen, H., Saberi, M., & Nizar, K. (2020). The Development of Parameter Estimation Method for Chinese Hamster Ovary Model using Black Widow Optimization Algorithm. *International Journal of Advanced Computer Science and Applications*, 11(11). <https://doi.org/10.14569/ijacsa.2020.0111126>
- [57] Naderi, S., Blondin, M. J., & Rezaie, B. (2022). Optimizing an adaptive fuzzy logic controller of a 3-DOF helicopter with a modified PSO algorithm. *International Journal of Dynamics and Control*, 11(4), 1895–1913. <https://doi.org/10.1007/s40435-022-01091-4>
- [58] Napiórkowski, J. J., Piotrowski, A., Karamuz, E., & Senbeta, T. B. (2022). Calibration of conceptual rainfall-runoff models by selected differential evolution and particle swarm optimization variants. *Acta Geophysica*, 71(5), 2325–2338. <https://doi.org/10.1007/s11600-022-00988-0>
- [59] Naser, M. Z., & Naser, A. (2024). The Firefighter Algorithm: A Hybrid Metaheuristic for Optimization Problems. *arXiv preprint*. <https://doi.org/10.48550/arxiv.2406.00528>
- [60] Nasr, M., Farouk, O., Mohamedeen, A., Elrafie, A., Bedeir, M., & Khaled, A. (2020). Benchmarking Meta-heuristic Optimization. *International Journal of Advanced Networking and Applications*, 11(6). <https://doi.org/10.35444/ijana.2020.11063>
- [61] Niroumandrad, N., Lahrichi, N., & Lodi, A. (2024). Learning tabu search algorithms: A scheduling application. *Computers & Operations Research*, 170, 106751. <https://doi.org/10.1016/j.cor.2024.106751>
- [62] Odeyemi, J., & Zhang, W. (2025). Benchmarking Randomized Optimization Algorithms on Binary, Permutation, and Combinatorial Problem Landscapes. <https://doi.org/10.48550/ARXIV.2501.17170>

- [63] Okechukwu, C., Šilhavý, R., Oyelere, S. S., & Šilhavý, P. (2024). Comparative Review of Single-Criteria and Multi-Criteria Optimisation Problems using Meta-heuristic Algorithms. *Research Square*. <https://doi.org/10.21203/rs.3.rs-4973948/v1>
- [64] Pacheco, J., Saiz-Vázquez, O., Casado, S., & Landa, S. U. (2023). A multistart tabu search-based method for feature selection in medical applications. *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-023-44437-4>
- [65] Pawłowski, J., Kutorasiński, K., & Szewczyk, M. (2022). Multifrequency nonlinear model of magnetic material with artificial intelligence optimization. *Scientific Reports*, 12(1). <https://doi.org/10.1038/s41598-022-23810-9>
- [66] Pelleau, M., Hentenryck, P. V., & Truchet, C. (2009). Sonet Network Design Problems. *arXiv preprint*, 5, 81–95. <https://doi.org/10.4204/eptcs.5.7>
- [67] Piotrowski, A., Napiórkowski, J. J., & Piotrowska, A. E. (2023). Particle Swarm Optimization or Differential Evolution—A comparison. *Engineering Applications of Artificial Intelligence*, 121, 106008. <https://doi.org/10.1016/j.engappai.2023.106008>
- [68] Rostami, M., Berahmand, K., & Forouzandeh, S. (2021). A novel community detection based genetic algorithm for feature selection. *Journal Of Big Data*, 8(1). <https://doi.org/10.1186/s40537-020-00398-3>
- [69] Sarhani, M., Voß, S., & Jovanović, R. (2022). Initialization of meta-heuristics: comprehensive review, critical analysis, and research directions. *International Transactions in Operational Research*, 30(6), 3361–3397. <https://doi.org/10.1111/itor.13237>
- [70] Schneider, P., Santiago, X. G., Soltwisch, V., Hammerschmidt, M., Burger, S., & Rockstuhl, C. (2019). Benchmarking Five Global Optimization Approaches for Nano-optical Shape Optimization and Parameter Reconstruction. *ACS Photonics*, 6(11), 2726–2733. <https://doi.org/10.1021/acsp Photonics.9b00706>
- [71] Shahmanzari, M., & Aksen, D. (2020). A Multi-Start Granular Skewed Variable Neighborhood Tabu Search for the Roaming Salesman Problem. *Applied Soft Computing*, 102, 107024. <https://doi.org/10.1016/j.asoc.2020.107024>
- [72] Sharifi-Noghabi, H., Mashhadi, H. R., & Shojaei, K. (2016). A novel mutation operator based on the union of fitness and design spaces information for Differential Evolution. *Soft Computing*, 21(22), 6555–6562. <https://doi.org/10.1007/s00500-016-2359-8>
- [73] Sinha, K., Chakraborty, S., Bardhan, A., Saha, R., Chakraborty, S., & Biswas, S. (2024). A New Differential Gene Expression Based Simulated Annealing for Solving Gene Selection Problem: A Case Study on

- Eosinophilic Esophagitis and Few Other Gastro-Intestinal Diseases. *Research Square*. <https://doi.org/10.21203/rs.3.rs-4997447/v1>
- [74] Sohail, M. S., Saeed, M. O. B., Rizvi, S. Z., Shoaib, M., & Sheikh, A. U. H. (2014). Low-Complexity Particle Swarm Optimization for Time-Critical Applications. *arXiv preprint*. <https://doi.org/10.48550/arXiv.1401.0546>
 - [75] Suman, B., & Kumar, P. (2005). A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, 57(10), 1143–1160. <https://doi.org/10.1057/palgrave.jors.2602068>
 - [76] Tang, J., & Wang, L. (2024). A whale optimization algorithm based on atom-like structure differential evolution for solving engineering design problems. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-023-51135-8>
 - [77] Tomar, V., Bansal, M., & Singh, P. (2024). Metaheuristic Algorithms for Optimization: A Brief Review. 238. <https://doi.org/10.3390/engproc2023059238>
 - [78] Utkarsh, & Jain, P. (2024). Predicting bentonite swelling pressure: optimized XGBoost versus neural networks. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-68038-x>
 - [79] Valadez-Vergara, R., & Szabó, N. P. (2024). Level of thermal maturity estimation in unconventional reservoirs using interval inversion and simulating annealing method. *Acta Geophysica*.
 - [80] Vert, D., Willsch, M., Yenilen, B., Sirdey, R., Louise, S., & Michielsen, K. (2024). *Benchmarking quantum annealing with maximum cardinality matching problems*. *Frontiers in Computer Science*, 6. <https://doi.org/10.3389/fcomp.2024.1286057>
 - [81] Wang, X., & Chen, Y. (2013). *Metaheuristic Optimization: Nature-Inspired Algorithms and Applications*.
 - [82] Wang, B., Liang, Y., Yuan, M., Zhang, H., & Liao, Q. (2019). A metaheuristic method for the multireturn-to-depot petrol truck routing problem with time windows. *Petroleum Science*, 16(3), 701–710. <https://doi.org/10.1007/s12182-019-0316-8>
 - [83] Wang, H., Deng, Y., Huangshui, H., Liu, K., & Xinji, F. (2024). Particle Swarm Optimization Based Mobile Sink Routing Protocol in Wireless Sensor Networks. *Research Square*. <https://doi.org/10.21203/rs.3.rs-3974420/v1>
 - [84] Wehrens, R., & Buydens, L. M. C. (2000). Classical and nonclassical optimization methods. In R. A. Meyers (Ed.), *Encyclopedia of Analytical Chemistry* (pp. 9678–9689). John Wiley & Sons Ltd.

- [85] Weiel, M., Götz, M., Klein, A., Coquelin, D., Floca, R., & Schug, A. (2021). Dynamic particle swarm optimization of biomolecular simulation parameters with flexible objective functions. *Nature Machine Intelligence*, 3(8), 727–735. <https://doi.org/10.1038/s42256-021-00366-3>
- [86] Yan, W., Li, N., & Zhang, X. (2023). Enhancing supply chain management in the physical internet: a hybrid SAGA approach. *Scientific Reports*, 13(1). <https://doi.org/10.1038/s41598-023-48384-y>
- [87] Yang, X. (2023). Nature-Inspired Optimization for Virtual Machine Allocation in Cloud Computing: Current Methods and Future Directions. *International Journal of Advanced Computer Science and Applications*, 14(11). <https://doi.org/10.14569/ijacsa.2023.01411121>
- [88] Yang, Z., & Burn, D. H. (2019). Automatic feature selection and weighting for the formation of homogeneous groups for regional IDF estimation. *Journal of Hydrology*, 575, 292–302. <https://doi.org/10.1016/j.jhydrol.2019.05.015>
- [89] Yuan, Q., & Yin, G. (2015). Analyzing convergence and rates of convergence of particle swarm optimization algorithms using stochastic approximation methods. *IEEE Transactions on Automatic Control*, 60(7), 1760–1773. <https://doi.org/10.1109/tac.2014.2381454>
- [90] Zambrano-Gutierrez, D. F., Cruz-Duarte, J. M., Avina-Cervantes, J. G., Ortíz-Bayliss, J. C., Yanez-Borjas, J. J., & Amaya, I. (2023). Automatic design of metaheuristics for practical engineering applications. *IEEE Access*, 11, 7262–7275. <https://doi.org/10.1109/access.2023.3236836>
- [91] Zhang, Z., Zhu, J., & Nie, F. (2024). A novel hybrid adaptive differential evolution for global optimization. *Scientific Reports*, 14(1). <https://doi.org/10.1038/s41598-024-70731-w>
- [92] Zhong, R., Cao, Y., Zhang, E., & Munetomo, M. (2024). Introducing Competitive Mechanism to Differential Evolution for Numerical Optimization. *arXiv preprint*, arXiv:2406.05436. <https://doi.org/10.48550/arxiv.2406.05436>