GÜNEYDOĞU ANADOLU, DOĞU ANADOLU VE NAHÇIVAN ARASINDA TİCARET YOLUNUN ACO ALGORİTMASI İLE OPTİMİZASYONU

Bahadur ALIZADA*

Özet: Bu makalede, Güneydoğu Anadolu, Doğu Anadolu ve Nahçıvan Özerk Cumhuriyeti arasında bir ticaret yolunun oluşturulması ve optimize edilmesi için bir yaklaşım sunulmuştur. Bu ticaret yolunun oluşturulması, bölgesel ekonomik işbirliğini artırmayı, ticaret hacmini artırmayı ve ekonomik kalkınmayı teşvik etmeyi amaçlamaktadır. Gezgin Satıcı Problemi (TSP) kullanılarak Karınca Aslanı Optimizasyonu (ACO) algoritması bu amaçla uygulanmıştır. ACO algoritması, karıncaların yiyecek kaynaklarını bulmak için kullandığı doğal bir davranışı modelleyen bir meta-sezgisel optimizasyon algoritmasıdır. Bu algoritma, Güneydoğu Anadolu, Doğu Anadolu ve Nahçıvan'daki büyük şehirler arasındaki en uygun ticaret yolunu bulmak için kullanılmıştır. Elde edilen ticaret yolu, maliyeti en aza indirirken etkinlik ve hızı artırabilir.

Anahtar Kelimeler: "*Güneydoğu Anadolu", "Doğu Anadolu", "Nahçıvan Özerk Cumhuriyeti", "Gezgin Satıcı Problemi" (TSP), "Karınca Aslanı Optimizasyonu" (ACO), "Bölgesel Ekonomik İşbirliği", "Ekonomik Kalkınma", "Meta-Sezgisel Optimizasyon Algoritmaları"*

Optimization of the Trade Route Between Southeastern Anatolia, Eastern Anatolia, and Nakhchivan with ACO Algorithm

Abstract: This article presents an approach for the establishment and optimization of a trade route between Southeastern Anatolia, Eastern Anatolia, and the Autonomous Republic of Nakhchivan. The aim of establishing this trade route is to enhance regional economic cooperation, increase trade volume, and promote economic development. The Ant Colony Optimization (ACO) algorithm is applied using the Traveling Salesman Problem (TSP) to achieve this goal. ACO algorithm, which models the natural behavior of ants in finding food sources, is utilized to find the most optimal trade route among the major cities in Southeastern Anatolia, Eastern Anatolia, and Nakhchivan. The resulting trade route can minimize costs while enhancing efficiency and speed.

Keywords: *"Southeastern Anatolia", "Eastern Anatolia", "Nakhchivan Autonomous Republic", "Traveling Salesman Problem" (TSP), "Ant Colony Optimization" (ACO), "Regional Economic Cooperation", "Economic Development", "Meta-Heuristic Optimization Algorithms"*

Giriş

Güneydoğu Anadolu ve Doğu Anadolu bölgeleri ile Nahçıvan Özerk Cumhuriyeti arasındaki ticaret yolu, bölgesel ekonomik işbirliğinin artırılması, ticaretin geliştirilmesi ve ekonomik kalkınmanın teşvik edilmesi açısından büyük öneme sahiptir. Bu makalede, bu ticaret yolunun oluşturulması ve optimize edilmesi için bir yaklaşım sunulacaktır. Makalenin odak noktası, bu ticaret yolunun oluşturulması için Gezgin Satıcı Problemi (TSP) kullanılarak Karınca Aslanı Optimizasyonu (ACO) algoritmasının nasıl uygulandığıdır.

Güneydoğu Anadolu, Doğu Anadolu ve Nahçıvan Arasındaki Ticaret Potansiyeli:

Bu bölgeler arasında ticaretin artırılması, coğrafi yakınlık ve tarihsel bağlar gibi faktörlerle desteklenmektedir. Ancak, mevcut altyapı ve lojistik zorluklar, ticaretin etkin bir şekilde gerçekleştirilmesini engelleyebilir. Bu nedenle, bir ticaret yolunun oluşturulması ve optimize edilmesi gereklidir.

Gezgin Satıcı Problemi (TSP) ve Ticaret Yolu Optimizasyonu:

TSP, bir gezgin satıcının belirli bir dizi noktayı (şehirleri) ziyaret etmesi gereken en kısa yolun bulunması problemidir. Bu bağlamda, Güneydoğu Anadolu, Doğu Anadolu ve Nahçıvan'daki büyük şehirlerin belirli bir sırayla ziyaret edilmesi gereken en kısa ticaret yolunun bulunması da bir tür TSP olarak ele alınabilir.

Karınca Aslanı Optimizasyonu (ACO) Algoritması:

ACO, karıncaların yiyecek kaynaklarını bulmak için kullandığı doğal bir davranışı modelleyen bir meta-sezgisel optimizasyon algoritmasıdır. Karıncalar, feromonlar aracılığıyla en iyi yolun belirlenmesinde bilgi paylaşır ve bu bilgiyi kullanarak koloni olarak en uygun yol bulunur. Bu özellik, ticaret yolu optimizasyonunda da kullanılabilir.

* Öğretmen, Bakü Avrasya Üniversitesi, "Matematik ve BİT" Bölümü. e-mail: bahaduralizade@gmail.com

DOI :10.30546/100981.1.2024.0148

Ticaret Yolu Oluşturma ve Optimizasyonunda ACO'nun Uygulanması:

Güneydoğu Anadolu, Doğu Anadolu ve Nahçıvan'daki büyük şehirlerin ve ticaret merkezlerinin coğrafi konumları bir ağ oluşturur. ACO algoritması, bu şehirler arasındaki en uygun ticaret yolunu bulmak için kullanılabilir. Karıncaların feromonlarına benzetilen yapay feromonlar, bu yolun belirlenmesinde rehberlik eder.

Ticaret Yolunun Önemi:

Bu ticaret yolunun oluşturulması, bölgesel ekonomik işbirliğini artırabilir, ticaret hacmini artırabilir ve ekonomik kalkınmayı teşvik edebilir. Ayrıca, bu yol sayesinde lojistik maliyetler azaltılabilir ve ticaret süreçleri daha verimli hale getirilebilir.

1. Yapay Zeka Algoritmalarının Gezgin Satıcı Problemlerinde Kullanılması.

Gezgin Satıcı Problemi (GSP), matematiksel ve bilgisayar bilimlerinde önemli bir optimizasyon problemidir. Temel olarak, bir satıcının belirli bir dizi noktayı (genellikle şehirleri) ziyaret etmesi gereken en kısa veya en ekonomik yolun bulunması problemidir. Bu problem, hem bilgisayar bilimleri hem de operasyon araştırması alanlarında geniş bir uygulama alanına sahiptir ve birçok gerçek dünya senaryosunda karşılaşılır.

Gezgin Satıcı Probleminin önemi, pratik ticari ve lojistik uygulamalarda optimal rotaların belirlenmesi gerekliliğinden kaynaklanır. Örneğin, bir satıcının birden fazla şehri ziyaret etmesi gereken bir dağıtım ağı kurması veya bir kargo şirketinin farklı müşterilere paketler dağıtması gibi durumlarda GSP kullanılır. Ayrıca, telekomünikasyon ve ulaşım sektörlerinde de yaygın olarak karşılaşılır.

GSP, NP-zor bir problemdir, yani çözümü doğrusal bir zaman karmaşıklığına sahip değildir ve en iyi çözümü bulmak bilgisayarlar için hesaplama açısından maliyetli olabilir. Bu nedenle, GSP'nin pratikte kullanışlı çözümlerinin geliştirilmesi için farklı yaklaşımlar ve algoritmalar araştırılmıştır.

Birçok yaklaşım GSP'yi çözmek için geliştirilmiştir, bunlar arasında en popüler olanları; kesin çözümler sağlayan tam tarama yöntemleri, yaklaşık çözümler sağlayan sezgisel ve meta-sezgisel optimizasyon algoritmalarıdır. Sezgisel optimizasyon algoritmaları arasında en çok kullanılanlardan biri Karınca Kolonisi Optimizasyonu (Ant Colony Optimization - ACO) ve Simüle Edilmiş Tavlama (Simulated Annealing) gibi algoritmalardır.

GSP'nin önemi, birçok endüstride etkili ve verimli bir şekilde kullanılabilecek birçok problemi çözme kapasitesinden kaynaklanır. Doğru bir şekilde ele alındığında, GSP'nin çözümü, işletmelerin maliyetlerini düşürme, zamanı ve kaynakları etkin bir şekilde yönetme, lojistik süreçleri optimize etme ve müşteri memnuniyetini artırma potansiyeline sahiptir.

Sonuç olarak, Gezgin Satıcı Problemi, karmaşık ve pratik ticari sorunların çözümünde kritik bir rol oynar. GSP'nin çözümü, işletmelerin rekabet gücünü artırmalarına ve operasyonel verimliliklerini iyileştirmelerine yardımcı olabilir. Bu nedenle, GSP'ye yönelik araştırma ve geliştirme çabaları, ticari uygulamalarda sürekli olarak değerlendirilmelidir.

Farklı yapay zeka algoritmaları, doğal süreçlerden ve hayvan davranışlarından ilham alarak tasarlanmıştır ve çeşitli optimizasyon ve problem çözme görevlerinde kullanılırlar. Bu algoritmaların her biri belirli bir problem alanına, veri yapısına veya kullanım durumuna göre avantajlar ve dezavantajlar sunar. İşte ABC (Arı Kolonisi Optimizasyonu), WOA (Whale Optimization Algorithm), Dragonfly Algorithm ve ACO (Karınca Kolonisi Optimizasyonu) gibi farklı yapay zeka algoritmalarının uzunca açıklamaları:

ABC (Arı Kolonisi Optimizasyonu):

ABC, arıların yiyecek kaynaklarını bulmak için kullandıkları doğal davranışlarını modelleyen bir meta-sezgisel optimizasyon algoritmasıdır. Algoritma, yiyecek kaynaklarının keşfedilmesi ve sömürülmesi sürecini taklit eder. ABC, bir popülasyon arasında yiyecek kaynakları üzerinde rekabet eden bireylerin davranışlarını modelleyen üç temel bileşene sahiptir: bal arıları, yiyecek kaynakları ve dans iletişimi.

Bal arıları, çözüm adaylarını temsil ederken, yiyecek kaynakları potansiyel çözümleri temsil eder. Dans iletişimi, bulunan çözümlerin diğer bal arılarına iletilmesini sağlar. ABC'nin ana prensibi, belirli bir zamanda bir bal arısının, bir yiyecek kaynağına gitme olasılığının o yiyecek kaynağının kalitesi ile orantılı olduğudur.

ABC, genellikle karmaşık problemlerin optimizasyonunda başarılı olmuştur, özellikle değişken boyutlu ve çok boyutlu optimizasyon problemlerinde kullanılır.

WOA (Whale Optimization Algorithm):

WOA, balina sürülerinin avlanma davranışlarını taklit eden bir meta-sezgisel optimizasyon algoritmasıdır. Bu algoritma, bir balina sürüsünün avlanma stratejisini modelleyerek çözüm adaylarını optimize eder. WOA, bir lider balinanın yönlendirmesiyle balina sürüsünün avlanma sürecini simüle eder.

WOA'da, her bir balinanın konumu bir çözüm adayını temsil eder. Belirli bir iterasyonda, her balina, lider balinanın konumuna doğru hareket eder ve avlanma davranışını takip eder. Bu, çözüm adaylarını güncellemek için kullanılır. Ayrıca, WOA'da, her balinanın belirli bir hızı ve manevra yeteneği vardır, bu da algoritmanın çeşitliliğini artırır.

WOA, özellikle çok boyutlu, karışık tamsayılı ve sürekli optimizasyon problemlerinde etkilidir. Genellikle, doğal ortamda gözlemlenen davranışları taklit etme kabiliyeti sayesinde, WOA, karmaşık optimizasyon problemlerine çözüm sağlama potansiyeline sahiptir.

Dragonfly Algorithm:

Dragonfly Algorithm, doğal olarak sürüler halinde yaşayan ve avlanan yusufçukların davranışlarını modelleyen bir meta-sezgisel optimizasyon algoritmasıdır. Bu algoritma, yusufçukların avlanma stratejisini simüle ederek çözüm adaylarını optimize eder.

Dragonfly Algorithm, bir popülasyon içindeki bireylerin konumlarını ve hızlarını güncelleyen bir dizi kuralla çalışır. Bu kurallar, bireylerin birbirlerini takip etmelerini, avlanma alanlarını aramalarını ve avlanma davranışlarını sergilemelerini sağlar. Böylece, Dragonfly Algorithm, bir çözüm adayının kalitesini artırmak için bir arayış ve hareket stratejisi kullanır.

Bu algoritma, genellikle karmaşık, çok boyutlu ve genellikle değişken boyutlu optimizasyon problemlerinde kullanılır. Doğal dünyadan ilham alması nedeniyle, Dragonfly Algorithm, bazı optimizasyon problemlerinde diğer algoritmalardan daha etkili sonuçlar üretebilir.

ACO (Karınca Kolonisi Optimizasyonu):

ACO, karıncaların yiyecek kaynaklarını bulmak için kullandığı doğal bir davranışı modelleyen bir meta-sezgisel optimizasyon algoritmasıdır. ACO, bir koloni içindeki karıncaların feromon izlerini kullanarak en iyi yolu bulma yeteneğine dayanır.

ACO, bir problemi çözmek için bir karınca kolonisi oluşturur ve her bir karınca bir çözüm adayını temsil eder. Karıncalar, feromon izleri aracılığıyla belirli bir noktadan diğerine seyahat ederken iz bırakır. Bu feromon izleri, diğer karıncaların yol seçimlerini etkiler. Kısa yol izleyen karıncaların izleri güçlenirken, uzun yol izleyenlerin izleri zayıflar.

ACO, genellikle karmaşık, kombinasyonel optimizasyon problemlerinde etkilidir. Özellikle, gezgin satıcı problemi gibi birçok problemde başarılı sonuçlar elde edilmiştir. Doğal davranışı taklit etme kabiliyeti ve paralel hesaplama yetenekleri nedeniyle, ACO, birçok uygulamada tercih edilen bir optimizasyon yöntemidir.

Her biri benzersiz avantajlar ve dezavantajlara sahip olan ABC, WOA, Dragonfly Algorithm ve ACO gibi yapay zeka algoritmaları, farklı optimizasyon problemlerine uygun çözüm sağlama potansiyeline sahiptir. Bu algoritmaların doğal dünyadan ilham alması, karmaşık problemleri çözmek için etkili ve esnek bir yaklaşım sunar.

Karınca Aslanı Optimizasyonu (ACO), bu problemin çözümünde başarılı olan bir meta-sezgisel optimizasyon algoritmasıdır. İşte ACO algoritmasının temel prensiplerini içeren bir pseüdokodu:

1. Başlangıçta, bir karınca kolonisi oluşturulur. Her karınca bir şehirden başlar.
2. Karıncalar, feromon izleriyle belirli bir şehirden diğerine seyahat ederler.
3. Feromon izlerinin gücü, ziyaret edilen şehirlerdeki karıncalar tarafından belirlenir.
4. Her karınca, seçimlerini feromon konsantrasyonuna ve uzaklığa dayanarak yapar.
5. Karıncaların izlediği yolun uzunluğu hesaplanır.
6. Feromon izleri güncellenir. Kısa yol izleyen karıncaların izleri güçlenir, uzun yol izleyenlerin izleri zayıflar.
7. Belirli bir kriter sağlanana veya maksimum iterasyon sayısına ulaşılana kadar adımlar tekrarlanır.
8. En iyi yol, feromon izlerine dayanarak seçilir ve çözüm olarak sunulur.

Karınca Aslanı Optimizasyonu (Ant Colony Optimization - ACO), karıncaların yiyecek kaynaklarını bulmak için kullandığı doğal bir davranışı modelleyen bir meta-sezgisel optimizasyon algoritmasıdır. Bu algoritma, bir koloni içindeki karıncaların, feromon izleri aracılığıyla en uygun yolu bulma yeteneğine dayanır.

ACO algoritması, bir problemi çözmek için karıncaların davranışını taklit eder. Örneğin, gezgin satıcı problemi gibi bir problemin çözümünde kullanılabilir. Bu problemde, bir satıcının belirli bir dizi şehri ziyaret etmesi gereken en kısa yol bulunmaya çalışılır. ACO algoritması, bu şehirler arasındaki en uygun yolu bulmak için kullanılır.

Algoritmanın temel prensibi, karıncaların doğal davranışını taklit etmek ve feromon izlerini kullanarak en iyi yolu bulmaktır. İşte ACO algoritmasının genel adımları:

Koloninin Oluşturulması: Başlangıçta, bir karınca kolonisi oluşturulur. Her bir karınca, bir şehirden başlar.

Yol Seçimi: Her karınca, bir sonraki ziyaret edilecek şehri seçer. Bu seçim, feromon konsantrasyonu ve şehirler arasındaki mesafe gibi faktörlere dayanır.

Yol İzleme ve Feromon Güncelleme: Karıncalar, seçtikleri yolu izlerken, feromon izleri bırakırlar. Kısa yolları izleyen karıncaların izleri güçlenirken, uzun yolları izleyenlerin izleri zayıflar.

Feromon İzlerinin Güncellenmesi: Her bir iterasyon sonunda, feromon izleri güncellenir. Kısa yollar izleyen karıncaların bıraktığı feromonlar güçlenirken, uzun yollar izleyenlerin bıraktığı feromonlar zayıflar.

En İyi Yolun Seçilmesi: Tüm karıncalar yol izlerken, en iyi yol, feromon izlerine dayanarak belirlenir.

İterasyonların Tekrarlanması: Belirli bir kriter sağlanana veya maksimum iterasyon sayısına ulaşılana kadar adımlar tekrarlanır.

ACO algoritması, bir problemi çözmek için karıncaların kolektif davranışını ve feromon izlerini kullanır. Bu şekilde, karmaşık problemlerin çözümü için etkili bir optimizasyon yöntemi sağlar. Gezgin satıcı problemi gibi NP-zor problemler de dahil olmak üzere birçok problemde başarılı sonuçlar elde edilmiştir.

2. Türkiye Ve Nahçivan Arasında Ticaret Lojistiğinin Oluşturulması

Güneydoğu Anadolu, Doğu Anadolu ve Nahçıvan Özerk Cumhuriyeti arasında bir ticaret yolunun oluşturulması, bölgesel ekonomik işbirliğinin artırılması, ticaret hacminin artırılması ve ekonomik kalkınmanın teşvik edilmesi açısından büyük öneme sahiptir. Bu bölgedeki ülkeler ve bölgeler arasında ticaret potansiyeli yüksektir, ancak mevcut altyapı eksiklikleri ve lojistik zorluklar ticaretin etkin bir şekilde gerçekleşmesini engelleyebilir. Bu nedenle, bir ticaret yolunun oluşturulması ve optimize edilmesi, bölgesel kalkınma ve işbirliği için kritik öneme sahiptir.

Ticaret yolunun oluşturulması, coğrafi, ekonomik ve lojistik faktörlerin dikkate alınmasını gerektirir. İlk adım, bu bölgelerdeki büyük şehirlerin ve ticaret merkezlerinin belirlenmesidir. Güneydoğu Anadolu'da Gaziantep, Şanlıurfa, Diyarbakır gibi şehirler; Doğu

Anadolu'da Van, Erzurum, Elazığ gibi şehirler; Nahçıvan Özerk Cumhuriyeti'nde ise Nahçıvan şehri gibi noktalar bu ticaret yolunun ana durakları olabilir.

Daha sonra, belirlenen bu şehirler arasında doğrudan bir ticaret yolunun fizibilitesi değerlendirilmelidir. Bu değerlendirme, coğrafi engeller, ulaşım altyapısı, güvenlik durumu gibi faktörlerin analizini içermelidir. Olası ticaret yolları belirlendikten sonra, bu yolların maliyet analizleri yapılmalı ve en ekonomik ve etkili olanı seçilmelidir.

Ticaret yolunun oluşturulması ve optimize edilmesinde yapay zeka algoritmaları, özellikle Karınca Aslanı Optimizasyonu (ACO) gibi yöntemler, önemli bir rol oynayabilir. ACO algoritması, karmaşık problemleri çözmek için karıncaların doğal davranışını modelleyen bir meta-sezgisel optimizasyon algoritmasıdır. Bu algoritma, gezgin satıcı problemi gibi birçok problemin çözümünde başarıyla kullanılmıştır.

3. Kullanılan yöntem

Çalışmada kullanılan büyükşehirler :

- Erzurum
- Malatya
- Mardin
- Van
- Diyarbakır
- Gaziantep
- Şanlıurfa

Bu şehirler arasındaki mesafeleri hesaplamak için bu şehirlerin Google Maps üzerindeki koordinatlarını kullanabiliriz. Aşağıda bu şehirlerin koordinatları

№	Konumlar	X kordinatı	Y kordinatı
1	Nahçıvan	39.28509004878757	45.40193910525418
2	Erzurum	39.97473033583969	41.249085935133515
3	Malatya	38.50382339096735	38.31938910642071
4	Mardin	37.38152621267302	40.746213974898126
5	Van	38.54387521045952	43.373006700064344
6	Diyarbakır	37.99647660989878	40.20348542918557
7	Gaziantep	37.28271879000183	37.35359035108194
8	Şanlıurfa	37.29629551522495	38.757034469369714

Bu koordinatları kullanarak, her bir şehir çifti arasındaki doğrudan mesafeyi hesaplayabiliriz. Ancak, Google Maps koordinatları sıklıkla daha karmaşık bir koordinat sistemi kullanır (Derece, Dakika, Saniye - DMS veya Decimal Derece - DD gibi). Bu nedenle, verilen koordinatları doğru şekilde işlemek için gerekli dönüşümleri yapmamız gerekebilir. Örneğin, Decimal Derece koordinatlarını kullanırsak, Haversine formülünü kullanarak iki nokta arasındaki mesafeyi hesaplayabiliriz.

Mesafe matrisi için olası bir tablo aşağıdaki gibi olabilir (mesafeler kilometre cinsinden):

		Nahçıvan	Erzurum	Malatya	Mardin	Van	Diyarbakır	Gaziantep	Şanlıurfa
1	Nahçıvan	0	490	883	768	385	700	1008	875
2	Erzurum	490	0	411	396	374	315	623	491
3	Malatya	883	411	0	325	560	227	225	219
4	Mardin	768	396	325	0	417	91	348	200
5	Van	385	374	560	417	0	365	673	541

6	Diyarbakır	700	315	227	91	365	0	311	179
7	Gaziantep	1008	623	225	348	673	311	0	150
8	Şanlıurfa	875	491	219	200	541	179	150	

SONUÇ

ACO algoritması kullanılarak elde edilen ticaret yolu, maliyeti en aza indirirken etkinlik ve hızı artırabilir. Ancak, bu çabaların başarılı olması için ilgili tarafların işbirliği yapması ve gerekli altyapı yatırımlarının yapılması gerekmektedir. Ticaret yolu üzerindeki potansiyel engellerin belirlenmesi ve bunların üstesinden gelmek için stratejiler geliştirilmelidir.

Yapılan deneyler sonucunda en uygun yol aşağıdakı sıralamayla verilmiştir :

2317 - 1 5 6 4 8 7 3 2

Gelecek Çalışmalar:

Gelecekte, ticaret yolunun optimize edilmesi ve altyapı iyileştirmelerinin yapılması için daha fazla çalışma gerekmektedir. Ayrıca, ACO algoritmasının daha fazla iyileştirilmesi ve gerçek dünya koşullarına uyum sağlaması için çalışmalar yapılmalıdır. Bu şekilde, Güneydoğu Anadolu, Doğu Anadolu ve Nahçıvan arasındaki ticaretin geliştirilmesi ve bölgesel işbirliğinin artırılması için önemli adımlar atılabilir.

Gelecek çalışmalar için bir dizi öneri sunulabilir. Öncelikle, ticaret yolunun güvenliğini sağlamak ve sürdürülebilirliğini artırmak için çevresel etkilerin ve toplumun kabul edilebilirliğinin değerlendirilmesi gerekmektedir. Bunun yanı sıra, ticaret yolunun entegrasyonunu ve etkinliğini artırmak için dijital teknolojilerin kullanılması da incelenmelidir. Bu teknolojiler arasında izleme ve lojistik yönetimi için yapay zeka, nesnelerin interneti (IoT) ve blokzincir gibi yenilikçi çözümler bulunmaktadır.

Ayrıca, ticaret yolunun çevresel etkilerinin minimize edilmesi için sürdürülebilir ulaşım teknolojileri ve yeşil enerji kaynaklarına yönelik yatırımlar da önemlidir. Bu, hem ticaretin sürdürülebilirliğini sağlayacak hem de bölgesel ekonomik kalkınmayı destekleyecektir.

Bununla birlikte, ticaret yolunun kullanımını teşvik etmek için politika düzenlemeleri ve ticaretin kolaylaştırılması için mevzuatın geliştirilmesi de önemlidir. Yönetim ve işletme modelinin belirlenmesi ve ilgili taraflar arasında işbirliğinin güçlendirilmesi, ticaret yolunun etkin bir şekilde işletilmesini sağlayacaktır.

Son olarak, ticaret yolunun sosyal ve ekonomik etkilerinin izlenmesi ve değerlendirilmesi gerekmektedir. Bu, yolun etkinliğini değerlendirmek ve gelecekteki gelişmeler için yol haritası belirlemek açısından önemlidir.

Tüm bu önerilerin uygulanması, Güneydoğu Anadolu, Doğu Anadolu ve Nahçıvan arasında ticaretin geliştirilmesine ve bölgesel işbirliğinin artırılmasına önemli katkılar sağlayacaktır.

KAYNAKÇA

[1]. Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.

[2]. Çelik, M., Köylü, F., & Karaboğa, D., (2016). CoABCMiner: An Algorithm for Cooperative Rule Classification System Based on Artificial Bee Colony. INTERNATIONAL JOURNAL ON ARTIFICIAL INTELLIGENCE TOOLS , vol.25.

[3]. Danaci, M., Koylu, F., & Al-Sumaidaee, Z. A., (2021). Identification of Dynamic Models by Using Metaheuristic Algorithms. ADI Journal on Recent Innovation (AJRI) , vol.3, no.1, 36-58.

[4]. ALIZADA, B. (2020). Improved Whale Optimization Algorithm Based On π Number. International Scientific and Vocational Studies Journal, 4(1), 21-30.

[5]. J. H. Holland, “Genetic Algorithms,” Scientific American, vol. 267, no. 1, pp. 66 72, 1992.
[6]. Murata, T., & Ishibuchi, H. (1995, November). MOGA: multi-objective genetic algorithms. In IEEE international conference on evolutionary computation (Vol. 1, pp. 289-294). IEEE Piscataway.
[7]. Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: an overview. Soft computing, 22, 387-408.
[8]. Coello, C. C., & Lechuga, M. S. (2002, May). MOPSO: A proposal for multiple objective particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) (Vol. 2, pp. 1051-1056). IEEE.
[9]. Karaboga, D. (2010). Artificial bee colony algorithm. scholarpedia, 5(3), 6915.
[10]. Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. Applied mathematics and computation, 214(1), 108-132.
[11]. Akay, B., Karaboga, D., Gorkemli, B., & Kaya, E. (2021). A survey on the artificial bee colony algorithm variants for binary, integer and mixed integer programming problems. Applied Soft Computing, 106, 107351
[12]. Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. Swarm and Evolutionary Computation, 2, 39-52.
[13]. Hancer, E., Xue, B., Zhang, M., Karaboga, D., & Akay, B. (2015, May). A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information. In 2015 IEEE congress on evolutionary computation (CEC) (pp. 2420-2427). IEEE.
[14]. Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27(4), 1053-1073.
[15]. Ouafa, H., Linda, S., & Tarek, B. (2017, December). Multi-objective optimal power flow considering the fuel cost, emission, voltage deviation and power losses using Multi-Objective Dragonfly algorithm. In Proceedings of the international conference on recent advances in electrical systems (pp. 191-197).
[16]. Khalilpourazari, S., & Khalilpourazary, S. (2020). Optimization of time, cost and surface roughness in grinding process using a robust multi-objective dragonfly algorithm. Neural Computing and Applications, 32(8), 3987-3998.
[17]. Goodman, N. R. (1963). Statistical analysis based on a certain multivariate complex Gaussian distribution (an introduction). The Annals of mathematical statistics, 34(1), 152-177.
[18]. Arnold, B. C., & Beaver, R. J. (2000). The skew-Cauchy distribution. Statistics & probability letters, 49(3), 285-290.
[19]. Yang X-S (2010) Nature-inspired metaheuristic algorithms, 2nd edn. Luniver Press
[20]. Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation, 8(2), 173-195.
[21]. Van Veldhuizen, D. A. (1999). Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Air Force Institute of Technology.
[22]. ÇELİK, M., KARABOĞA, D., & KÖYLÜ, F., (2011). Artificial bee colony data miner (ABC-Miner) . 2011 International Symposium on INnovations in Intelligent SysTems and Applications, INISTA 2011 (pp.96-100). Istanbul-Kadikoy, Turkey
[23]. KÖYLÜ, F., (2017). Online ABC miner: An online rule learning algorithm based on Artificial Bee Colony algorithm . 8th International Conference on Information Technology, ICIT 2017 (pp.653-657). Amman, Jordan

[24]. ALIZADA, B. A Novel Hybrid Algorithm: Sine Cosine Harmony Search Algorithm for Global.
[25]. Hancer, E., Xue, B., Zhang, M., Karaboga, D., & Akay, B. (2018). Pareto front feature selection based on artificial bee colony optimization. Information Sciences, 422, 462-479.