

# Dağıtılmış Etmen Tabanlı Karınca Kolonisi Optimizasyonu Sistem Tasarımı

Musa ÜNSAL<sup>1</sup>, Özgür Koray ŞAHİNGÖZ<sup>2</sup>

<sup>1</sup> Havaçılık ve Uzay Teknolojileri Enstitüsü, Bilgisayar Mühendisliği Bölümü, İstanbul

<sup>2</sup> Hava Harp Okulu, Bilgisayar Mühendisliği Bölümü, İstanbul

munsal1@hho.edu.tr, o.sahingoz@hho.edu.tr

**Özet:** Büyük boyutlu uygulamaların paralel veyada dağıtılmış yapıda tasarlanması işlemi, uzman yazılımcılar tarafından işlem gücünün artırılması ve daha hızlı sonuca ulaşılması açısından tercih edilmektedir. Bu yaklaşımlarda bileşenler arası koordinasyon ve iletişim açısından ciddi sorunlar karşımıza çıkarmaktadır. Bu nedenle sanal ortamda buna benzer problemlerin çözümünde gerçek dünyada başarı sağlamış olan doğa esinlenimli yöntemlerden faydalanılmaktadır. Bu yöntemlerde gerçek hayattaki grup bireyleri arasındaki koordinasyon ve haberleşme, sanal ortama transfer edilmekte ve bu sayede doğal evrim sürecinin, sanal evrim sürecine dönüştürülmesi sağlanmaktadır.

Bu bildiri doğa esinlenimli yöntemlerden olan Karınca Kolonisi Optimizasyonu algoritmasının dağıtılmış yapıda çoklu etmenlerin kullanımı ile çözülmesini anlatmaktadır. Bu sistemde her bir karıncanın bir hareketli etmen olarak tanımlanması amaçlanmaktadır. Karıncalar, düğümler (statik etmenler) arasında belirli bir karar verme mekanizmasına ve rasgeleliğe göre hareket etmekte ve geçtikleri yol üzerine bir feromon bırakmaktadırlar. Bu feromon sayesinde her bir karınca diğer karıncalarla dolaylı olarak haberleşmektedir. Geçen zaman ile bırakılan feromon buharlaşmakta ve ağırlığı azalmaktadır. Yol üzerinde kalan fazla feromon miktarı bir sonraki karıncanın aynı yolu izleme olasılığını artırmaktadır. Bu bildiride Karınca Kolonisi Optimizasyonu algoritması dağıtılmış ortamda bir çoklu etmen uygulaması olarak tasarlanmış ve eniyileme algoritmasının sadece hareketli etmen içerisinde olmasının sağlanması ile sisteme geniş bir esneklik sağlanmıştır.

**Anahtar Sözcükler:** Karınca Koloni Optimizasyonu, Gezgin Satıcı Problemi, Eniyileme, Etmen Tabanlı Sistemler, İnternet.

## Distributed Agent-Based Ant Colony Optimization System Design

**Abstract:** The process of designing large-scale applications in parallel and distributed manner is preferred by expert software programmers to increase the systems processing power and to reach results quickly. This approach raises serious problems in terms of coordination and communication. In artificial environment, to solve this type of problems natural inspired methods, which have proved its success in real world, is preferred. Coordination and communication mechanism between group members is transferred from real world to the artificial environment, and as a result of this, the natural evolution process is transferred to the virtual evolution process.

This paper presents how to solve natural-inspired ant colony optimization algorithm by using distributed multi-agents structure. In this system, each ant is designed as a mobile agent. Ants move between nodes (static agents) according to a specific decision-making and random

selection rule, and during the trip, the ants leave a chemical pheromone trail on the ground. An ant indirectly communicates with other ants with this pheromone. The pheromone evaporates and its weight is lessened by the time. The more pheromone remaining on the path increases the next ant's selection probability to the same path. In this paper ant colony optimization algorithm designed as a distributed multi-agent implementation, and providing the optimization algorithm in mobile-agent makes system more flexible.

**Keywords:** Ant Colony Optimization, Travelling Salesman Problem, Optimization, Agent-Based Systems, Internet.

## 1. GİRİŞ

Karmaşık problemlerin çözümünde genellikle yüksek işlem gücüne ihtiyaç duyulmaktadır. Yüksek işlemgücü elde etmek içinde ya süper bilgisayarların kullanılması ya da dağıtılmış ortamda çok sayıda makine üzerinde işlem yapılması gerekmektedir. Her iki durumda da farklı işlemciler üzerinde çalışması gereken farklı süreçlerin tanımlanması gerekmektedir. Grup halinde çalışan bu süreçlerin birbirleri ile etkin bir koordinasyonunun sağlanması bilgisayar haberleşmesindeki temel problemlerdendir.

Bir çok farklı problemde olduğu gibi, bu tür sorunların halledilmesi için gerçek hayattaki canlıların kullandıkları yöntemlerden faydalanılabilmektedir. Bilgisayar dünyasında Doğa Esinlenimli Yöntem(DEY) olarak adlandırılan bu tip yöntemlerde grup koordinasyonunun ve haberleşmesinin nasıl sağlandığı incelenmekte ve gerçek hayattaki evrim sürecinin sanal ortama taşınması ile sanal varlıkların(programların) kendilerini geliştirmeleri ve çözüme daha etkin bir şekilde ulaşmaları sağlanmaktadır.

DEY'ler arasında üzerinde en çok çalışılan algoritmalarından birisi de Karınca Kolonisi Optimizasyonu(KKO) dur. KKO genellikle doğal karıncaların çevrelerini yiyecek için araştırırken yaptıkları hareketlerden esinlenerek oluşturulmuştur. Arama sonrası yuvaya geri dönüş yolunda karıncalar yol güzergâhına feromon adı verilen kimyasal bir madde salgırlarlar. Yuvada bulunan diğer karıncalar yola bırakılan bu kimyasal

feromon maddesini kullanarak yiyeceğe ulaşırlar. Yuva ile yiyecek arasında oluşturulan yollardan hangisinde daha çok feromon maddesi varsa o yolun seçilme olasılığı artmaktadır. Yol güzergâhına bırakılan feromon maddesine güncelleme ve buharlaşma özellikleri katılarak bu maddenin eniyi sonuca ulaşmada yardımcı bir etken olması sağlanmıştır.

Sanal ortamdaki KKO uygulamalarında düğümler arasındaki yolların uzunlukları feromon maddesinin buharlaşması ve güncellenmesi işlemlerinin uygun bir şekilde yapılması gerekmektedir. Düğümler arasında bulunan alternatif yollar arasında uzun olan yollardaki feromon miktarı zaman geçtikçe daha fazla azalmakta ve bu yolların seçilme olasılığı karıncaların hareketine bağlı olarak tekrarlar sonrasında sifira indirgenmektedir. Fakat aynı şekilde kısa yollardaki feromon miktarı ise zaman içerisinde yine karıncaların hareketine ve kullanılma sıklığına bağlı olarak artmakta ve seçilme olasılığı da yükselmektedir. Sonuçta kısa yolların yenilenme ve güncellenmesi daha sık yapılmakta ve böylelikle sonuca daha çabuk ulaşılmasına olanak sağlamaktadır. Böylece karıncalar sezgisel olarak feromon maddesini kullanarak gidecekleri noktaya en kısa yoldan ulaşmayı başarmaktadır [1].

Yapmış olduğumuz araştırmalar dağıtık KKO ile ilgili yapılan çalışmaların sezgisel ve basit arama yöntemlerini desteklemediğini göstermektedir[1,2]. Aksine mevcut yaklaşımlar paralel ve yüksek performanslı bilgisayar mimarileri için KKO'nun basit

dağıtık yapısını dikkate almadan klasik sıralama algoritmalarının uyarlanması üzerine kurulmuştur.

Bu kapsamda KKO'nun çoklu etmen sistemler arakatmanı aracılığıyla dağıtık mimari üzerinde basit ve sezgisel yöntemlere dayandırılarak bir çoklu etmen teknolojisi üzerinde gerçekleştirilmesi hem gerçek hayat uygulamalarına daha fazla benzeyecek hem de daha verimli olacaktır. Bu araştırmanın amacı daha etkin bir hesaplama ve programlama ile KKO algoritmasını farklı bir bakış açısıyla çoklu etmen sistemleri üzerinde uygulamaktır. Tasarlanan Sistemin bu alanda etkin kullanılan bir problem ortamı olan Gezgin Satıcı Problemi üzerinde test edilmesi amaçlanmaktadır

Bildirinin devam eden bölümleri şu şekilde organize edilmiştir; Bölüm 2'de Karınca Kolonisi Optimizasyonu, Bölüm 3'de Gezgin Satıcı Problemleri ve Bölüm 4'te Etmen tabanlı teknolojilerin getirdikleri anlatılmıştır. Bölüm 5 ve Bölüm 6'da sırasıyla Oluşturulan Sistemin Yapısı ve Konuyla İlgili Çalışmalar anlatılmıştır. Bölüm 7'de Sonuçlar ve Gelecek Çalışmalar vurgulanarak bildiri sonlandırılmıştır.

## **2. KARINCA KOLONİSİ OPTİMİZASYONU**

Karınca koloni optimizasyon algoritmasını Marco Dorigo 1992 yılında İtalya'da yapmış olduğu tezinde ortaya atmış ve bu konu sonraki yıllarda başta Dorigo ve arkadaşları olmak üzere birçok bilim adamı tarafından geliştirilerek günümüz teknolojik dünyasına aktarılmıştır [3]. Halen konuyla ilgili birçok alanda çalışmalar devam etmektedir.

Bilgisayar mühendisliğinde böceklerin toplu hareketlerini optimizasyon ve kontrol algoritmalarına uygulanması önemli bir çalışma alanı olmuştur. Yapay karıncaların, böceklerin bu hareketini nasıl uygulayacağı ve bununla ilgili nasıl sonuçlara ulaşacağı da artık günümüzün vazgeçilmez çalışma konuları arasına girmiştir [5].

KKO algoritmaları sezgisel metotla çözülen ve gerçek karıncaların en kısa yolu bularak hedeften yuvalarına ulaşmalarını sağlayan hesaplamaları yaparlar. Karınca koloni optimizasyon algoritmaları çok sayıda yapay karıncayla oluşturulur ve çözümlü oluştururken karıncaların devamlı ve tekrarlı hareketleri sürdürmeleri gerekir. Bu döngü esnasında her yapay karınca kendi çözümünü oluşturur.

KKO algoritmaları bütünlük optimizasyon problemlerini çözerken koloni içerisindeki karıncaların etkileşiminden faydalanır [6]. Bu etkileşim feromon adı verilen bir kimyasal madde sayesinde olur. Karıncalar yiyecek ararken kendi yuvalarının etrafını araştırırlar. Yiyecek kaynağına ulaşan karınca bu yiyecek kaynağını belli kriterler çerçevesinde inceler ve buradan aldığı bir miktar yiyeceği yuvasına taşır. Karınca yiyeceği yuvasına taşırken dönüş yoluna feromon adı verilen kimyasal maddeyi salgılar. Yuvada bulunan diğer karıncalar bu feromon salgısı aracılığıyla en uygun yol güzergâhını bulmaya çalışırlar. Bütün karıncalar aynı metotla yiyecek arama ve tekrar yuvaya taşıma görevini yerine getirirler. Oluşturulan bu tekrarlı yol izleme faaliyeti sonrasında gidilecek rotalar üzerinde feromon miktarı zamanla değişir. Karıncaların sık kullandıkları yollar üzerinde biriken feromon miktarı artarken karıncaların seyrek kullandıkları ya da hiç kullanmadıkları yollar üzerinde ki feromon miktarı azalır veya sıfırlanır. Bu işlemde feromon güncelleme ve buharlaşma metoduyla algoritmaya uygulanır. Feromon güncelleme kullanılan yollar üzerine belli kurallar dâhilinde feromon eklenmesi yöntemiyle sağlanırken feromon buharlaşması kullanılmayan yollar üzerinde bulunan feromon miktarının azaltılması sayesinde gerçekleşir [1].

Her karınca rotasını oluştururken noktalar arasında hareket eder. Karıncanın rota oluşturma esnasında aşağıdaki üç işlemi uygulaması gerekmektedir [1];

- Belirli kriterlere göre hareketi başlatacak karınca için bir başlangıç noktası seç,

- Her noktayı ekleyerek bütün noktalara uğrayıncaya kadar feromon ve sezgisel matris verilerini kullanarak bir rota oluştur,
- İlk harekete başladığın noktaya tekrar geri dön.

Bütün noktalar dolaşılıp tur oluşturulduktan sonra karıncalar tur üzerindeki yollara feromon bırakabilirler. Feromon bırakma öncesinde karıncalar bu yolları yerel arama yöntemleriyle daha da geliştirebilirler. Şekil 1 de gösterilen algoritma KKO'nun statik bütünleşik optimizasyon problemlerine örnek bir algoritmadır.

```

FonksiyonKarıncaKoloniOptimizasyonu() {
  Parametreleriayarla,
  Feromondeğerleriniayarla
  Koşulsatılanıncayakadar{
  Karıncaçözümünüoluştur;
  Yerelaramaışleminitamamla (istenirse);
  Feromongüncellemesiyap;
  }
}

```

Şekil 1. Örnek KKO Algoritması

Bu algoritmada feromon güncellemesi bölümü hem feromon güncellemesini hem de feromon buharlaşmasını içermektedir.

KKO ilk olarak gezgin satıcı problemlerinde test edilmiştir. Gezgin satıcı problemleri literatürde çok araştırılmış ve incelenmiş problemlerdir. Bu çalışmada KKO gezgin satıcı problemleri üzerinden incelenecektir.

### 3. GEZGİN SATICI PROBLEMLERİ

Gezgin Satıcı Problemleri (GSP) satıcının evinden çıkıp verilen müşteri adreslerinin herbirine bir kez uğrayıp tekrar çıkış noktasına döndüğünde gittiği mesafenin en kısa olması gerektiği prensibiyle sonuca ulaşır. Genelde gezgin satıcı problemleri  $GSP=(S,Y)$  şeklinde gösterilir.

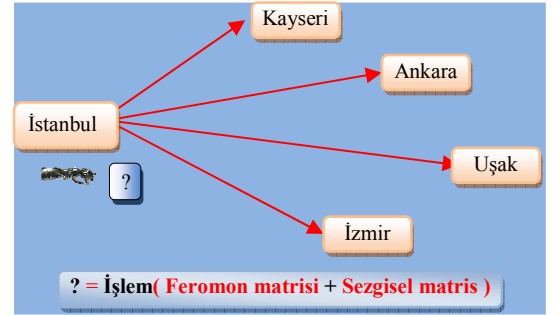
$$GSP=(S,Y)$$

S; Şehir sayısı (kümesi)

Y; Şehirlerarası bağlantı yolu (kümesi)

Her  $(i,j)$  yolu Y'nin bir elemanıdır ve  $d_{ij}$  ile gösterilen yol uzunluğuna sahiptir. Bu uzunluk S'nin elemanı olan i ve j arasındaki uzaklığı ifade eder [1].

GSP'lerinin çözümünde iki tane matrise ihtiyaç duyulur. Bunlar *feromon matrisi* ve *sezgisel matrislerdir* [7]. Şekil 2.'de Feromon ve sezgisel matris kullanılışı gösterilmektedir.



Şekil 2. Feromon ve sezgisel matrisin kullanılması

GSP'leri için uygun olan KKO'larında feromon matrisi feromon izleri ile ilişkilidir. Karıncaların geçiş güzergâhında bıraktığı feromonların tutulduğu matrisi ifade eder. Örneğin  $\tau_{ij}$ ; karıncanın i şehirden sonra j şehrine uğraması için geçerli olan kaydedilmiş feromon miktarını gösterir [1].

Sezgisel matris ise sezgisel yöntemlerde sezgisel veriyi ifade eder. i şehirden j şehrine hesaplanacak seçim olasılığı için gerekli olan veri bu matriste tutulur. Örneğin  $n_{ij} = 1/d_{ij}$  ise karıncanın i şehirden j şehrine uğraması olasılığı; i şehri ile j şehri arasındaki uzaklıkla ters orantılıdır. Bazı durumlarda i ve j şehirleri arasındaki uzaklık "0" olabilir. Bu durumda  $n_{ij}$  çok küçük bir değer olarak tanımlanır [1].

Sezgisel olasılık matrisi şehirlerarası uzaklıklar problem sonuca ulaşana kadar değişmeyeceği için sabit kalır. Burada amaç

karıncanın yerel eniyilemelere takılıp genel eniyilemeleri bulamamasını önlemektir.

Bugüne kadar gezgin satıcı problemleri için birçok karınca koloni algoritması kullanılmıştır. Bunlar arasındaki tek fark feromon güncellemesi ve feromonun yönetilmesiyle ilgili detaylardır. Tablo 1’de KKO’nun gezgin satıcı problemleri için kullanılan değişik versiyonlarının tarihi gelişimi ve isimleri görülmektedir [4].

Karınca sistemi ile 1997 yılında Dorigo ve Gambardella tarafından geliştirilen karınca koloni sisteminin benzerlikleri ve farklılıkları [12,16] aşağıda sıralanmıştır.

- Karınca sistemi ve karınca koloni sistemi seçim aşamasında öncelikle feromonu yüksek olan yolu seçer eğer yollardaki feromon eşitse rastgele bir yol seçimine izin verir.
- Karınca sisteminde feromon güncellemesi esnasında bütün yollarda, yol uzunluklarının tersi ile orantılı miktarda feromon yollara bırakılır ve sabit bir miktarda buharlaştırılır. Karınca koloni sisteminde ise sadece o anda tanımlanan yollardaki en kısa yola ait olan yollara feromon bırakılır ve buharlaşma sadece bu yollara uygulanır.
- Karınca sistemi 30 düğüm kadar çok iyi sonuç verir, 70 ve üzerindeki düğümlerde

iyi sonuçlardan uzaklaşır. Karınca koloni sistemi ise 100 düğüm ve üzerindeki büyük problemlerde iyi sonuç verir.

- Karınca sisteminde çözümlenen düğüm sayısına göre başarı sınırlı olabilmektedir. Karınca koloni sistemi ise yüzlerce düğüm kadar oldukça yüksek bir başarı sağlamaktadır.
- Karınca sistemi arama için genelde hareket seçim kuralını kullanırken karınca koloni sistemi karıncaların oluşturdukları rotaları dikkate alarak onların hareketinden de çıkarımda bulunurlar.
- Karınca koloni sisteminde feromon güncellemesiyle alternatif yolların seçilmesi olasılığı artırılır.

#### 4. ETMEN TABANLI TEKNOLOJİLERİN GETİRDİKLERİ

Yazılım dillerinin geçmişine bakıldığında, birinci nesil programlama dilleri, makine dilleri (ARM, Sun SPARC vb.), ikinci nesil çevirici diller (BAL (Basic Assembler), TASM (Turbo Assembler, Borland) vb.), yapısal diller (C, C++, Java vb.), çözüme yönelik diller (SQL, PL/SQL, MATLAB vb.) gibi birçok programlama dili kullanılmıştır.

Farklı bir bakış açısıyla yazılım dillerinin tarihi gelişimi kendi arasında gruplandırılarak

<i>Algoritma Adı</i>	<i>Yazan Kişi/Kişiler</i>	<i>Referans</i>
<i>Ant System (AS)</i>	Dorigo, Maniezzo, Colorni	[8]
<i>Elitist AS</i>	Dorigo	[1]
	Dorigo, Maniezzo, Colorni	[9]
<i>Ant-Q</i>	Gambardella, Dorigo	[10]
	Dorigo, Gambardella	[11]
<i>Ant ColonySystem</i>	Dorigo, Gambardella	[12]
<i>MAX-MIN AS</i>	Stutzle, Hoos	[13]
<i>Rank-Based AS</i>	Bullnheimer, Hartl, Strauss	[14]
<i>Hyper-Cube Framework (HCF)</i>	Blum, Dorigo	[15]

**Tablo 1.** Karınca koloni optimizasyonunun gezgin satıcı problemlerine uygulanan değişik versiyonlarının tarihi gelişimi ve isimleri

oluşturulmuş sıralaması Tablo 2'de gösterilmektedir.

<b>Programlama Dili</b>	<b>Periyod</b>
<b>Birinci Nesil Programlama Dilleri</b>	1940-1950 arası
<b>Makine Dili</b>	1950 'li yıllardan sonra
<b>Yüksek Seviyeli Diller</b>	1960 'li yıllardan sonra
<b>Çok Yüksek Seviyeli Diller</b>	1970 'li yıllardan sonra
<b>Yapay Zekâya Yönelik Diller</b>	1980 'li yıllardan sonra

**Tablo 2.**Yazılım dilleri tarihi gelişimi

Bilgisayar dünyasında kullanılan basit nesnelere artık kullanılabilirliğini kaybetmektedir. Oysaki etmenlerin sistem bileşenlerinden ayrıldığı nokta öncelikle akıllı olmalarıdır.

Son zamanlarda gelişen teknolojinde etkisiyle etmen tabanlı yazılımların kullanılması kaçınılmaz hale gelmiştir. Bunun sonucu olarak etmen tabanlı programların özellikle dağıtılmış, hareketli ve akıllı sistemlerin geliştirilmesinde etkin rol oynadığı görülmektedir.

Günümüz teknolojik dünyasında bilgisayarlar yalnızca basit işleri yapan makineler olmaktan çıkmış günlük hayatımızda ve kolaylaştırıcı işlerin planlanmasında ilk başvurulacak yardımcı olmuştur. Bilgisayarlar kendilerine verilen bu görevleri yerine getirebilmek için etmen tabanlı sistemlerin davranış gösterebilme yetkisine sahip olmaları ve çevre ile olan kimi zaman basit, kimi zaman da karmaşık yapıdaki özelliklerini kullanırlar. Bu sistemlerin daha etkin kullanılabilmesi için yapay zekâ gibi bazı yetenekler bu sistemlere kazandırılmış ve daha nitelikli düşünebilen sistemler geliştirilmiştir. Artık birbiriyle iletişimde olan sistemler ve programlar ön plana çıkmıştır. Bütün programlar ve sistemler her işi yapabilecek fonksiyonlara sahip olmaktan öte her program kendine özgü faaliyeti yerine getirir. Bu üretilen iş aynı zamanda başka sistem ve programlara da servis sağlayıcı olarak hizmet verir. Böylece her program ya

da sistem istediği her fonksiyonun sonucuna sadece servisi çağırarak ulaşabilir. Böylece aynı yer ve zamanda üretilmiş olmasalar da farklı veya ortak hedefler için farklı zamanlarda farklı yerlerde olabilirler. Bunun için gerekli şartlar koordinasyon ve iletişimdir. Bu koordinasyon ve iletişim etmenler tarafından kullanılır ve yapılacak işler etmenler tarafından yerine getirilir [17].

Etmen, sistem tasarımcısı tarafından bir ya da daha fazla işlemi yerine getirmek üzere görevlendirilen ve bu doğrultuda belirli bir noktaya kadar otonomi yüklenen sanal birimlerdir [18].

Wooldridge ve Jennings akıllı etmenleri kendi hedeflerini başarabilmek için esnek ve otonomik hareket edebilen bilgisayar bileşenleri olarak tanımlamaktadır [19]. Esnek özelliği ile bilgisayar bileşeninin asıl içermesi gereken özellikler aşağıdadır.

**Cevap veren (responsive) :** Etmen kendi etrafında oluşan tepkileri, kullanıcıyı, etmenleri, interneti vb. olayları algılayabilmeli ve buna karşılık kendi oluşturacağı tepkiyi verebilmelidir.

**Proaktif (proactive) :** Zeki bir etmen hedef odaklı hareket etmelidir. Bunun için de fırsatları değerlendirmeli, seçenekler öngörebilmeli ve hedefe ulaşmak için gereken tüm koşulları yerine getirerek inisiyatif kullanabilecek kabiliyette olmalıdır.

**Sosyal (social) :** Hedeflere ulaşabilmek için zeki etmenin başka sistem etmenleriyle işbirliği içinde olması gerekir. Gerçek dünyada nasıl ki insanlar birbirleriyle iletişim halinde bir işi yerine getiriyorlarsa, zeki etmenler de aynı şekilde birlikte çalışabilmelidirler.

Etmen ve çoklu etmenler arasındaki iletişim belli kurallara dâhilinde yapılmaktadır. IEEE etmenler ve çoklu etmen sistemler için 8 Haziran 2005 tarihinde Fiziksel ve Zeki Etmenler Kuruluşunu (FIPA-Foundation for Physical and Intelligent Agents) on birinci standartlar komitesi olarak kabul etmiştir. Bu kuruluş etmenlerin ve etmen teknolojilerinin

karşılıklı çalışabilirlik standartlarını belirlemektedir [20].

Etmenler son zamanlarda birçok gerçek dünya problemini özellikle insan zekâsını kullanabilecek özelliklerde bilgisayar sistem ve programlarıyla çözmektedirler. Benzer şekilde etmenlerin karınca koloni optimizasyonlarını kullanarak seyyar satıcı problemlerini çözmeye kullanılması da kaçınılmazdır. Geçmişten günümüze kadar farklı sistem tasarımlarıyla çözülmüş uygulamalar vardır. Bazı uygulamalarda şehirler etmen olarak planlanırken bazılarında hem karınca hem de uğranacak şehirler etmen olarak planlanmıştır [21,22].

## 5. OLUŞTURULAN SİSTEMİN YAPISI

Dağıtılmış KKO Sistemi JADE Etmen mimarisi üzerinde tasarlanmıştır. Sistem hareketli ve statik etmenlerden oluşmaktadır. Hareketli etmenler KKO sistemindeki karıncaları ifade etmekte olup, statik etmenler ise gezinti yapılacak noktaları/düğümüleri

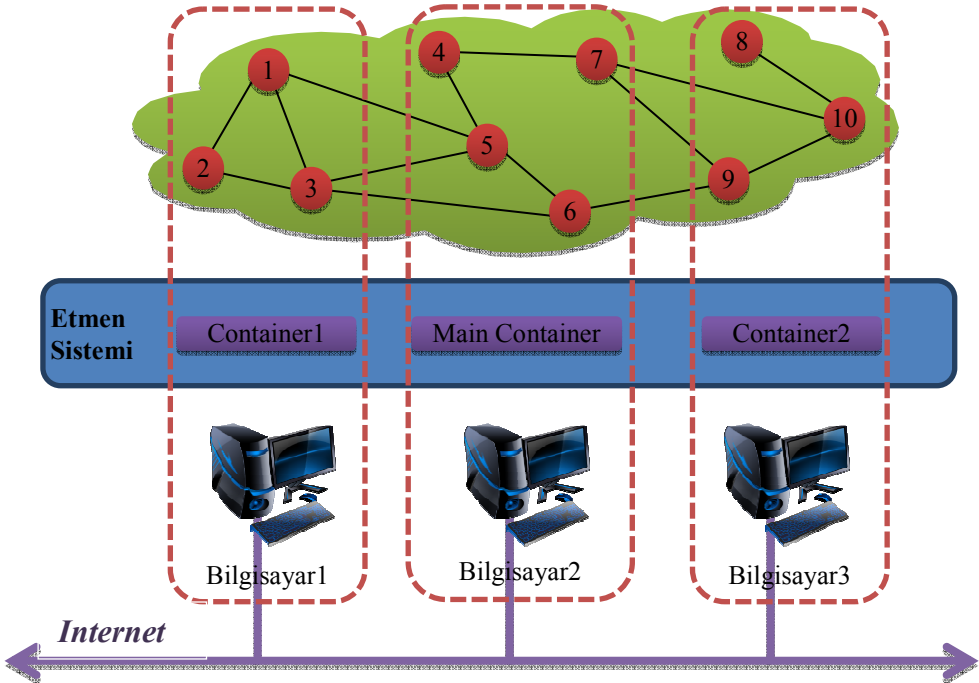
ifade etmektedir. Sistemin temel altyapısı Şekil 3'te gösterilmiş olup sistem çok sayıda etmenin farklı makineler üzerinde aynı anda çalıştırılmasına olanak sağlayacak şekilde tasarlanmıştır.

### 5.1. JADE Etmen Geliştirme Sistemi

JADE (Java Agent Development Framework) kâr gütmeyen bir organizasyon olan FIPA (Foundation for Intelligent Physical Agents) spesifikasyonlarına uygun bir çoklu etmen geliştirme sistemidir.

JADE Etmen platformu dağıtılmış ortamda etkileşime olanak sağlayan etmenlerin Java programlama dili ile geliştirilmesine olanak sağlamakta, bunun için ihtiyaç duyulan temel etmen yönetim sistemlerini ( adlandırma, serileştirme, göç etme, haberleşme vb.) kendi içinde barındırmaktadır.

Bir JADE etmen sisteminin oluşturulması için bir makinenin Main Container olarak tanımlanması ve diğer bilgisayarlarında kendi JADE Containerlarını oluşturarak bu ana makineye bağlanmaları gerekmektedir. Daha sonra etmenler arası haberleşme farklı



Şekil 3. Tasarlanan Sistemin Alt Yapısı

makinelere olmalarından bağımsız olarak tek bir makine üzerindeymiş gibi kolaylıkla yapılabilmektedir. JADE sistemi farklı etmen sistemleri arasında (ayrı Main Containerları olan) haberleşme olanağı da sağlamaktadır.

### 5.2. Düğüm Yapısı ve Statik Etmenler

Her bir düğüm statik etmen yapısında tasarlanmış olup kendi içerisinde şu verileri tutmaktadır.

- Temel tanımlayıcı ve etmen bilgileri
- Komşu düğümlerin listesi: Bu düğümler aynı makinede olabileceği gibi farklı makineler üzerinde de olabilir.
- Komşu düğümlerle aradaki bağlantıdaki feromon miktarı.
- Kendine gelen karıncalara (hareketli etmene) ait bilgiler

Statik etmenlerin bulunduğu makineye gelen hareketli etmenlerle koordinasyonları JADE sisteminin sağladığı FIPA-ACL (Agent Communication Language) etmen haberleşme dili ile sağlanır. Bu sayede ilgili düğüme gelen karınca düğümden gerekli bilgileri almış olur ve aynı zamanda düğüm üzerindeki feromon miktarında güncelleme yapar.

### 5.3. Karınca Yapısı ve Mobil Etmenler

Karınca yapısı sistem üzerinde hareketli etmen olarak tasarlanmıştır. Her bir karınca kendi içerisinde şu bilgileri barındırır;

- Temel tanımlayıcı ve etmen bilgileri
- Gidilebilecek komşu düğüm listesi
- Komşu düğümlerin masrafları ve feromon değerleri
- Dolaşmış olduğu düğüm listesi (dolaşma sırası ile)
- Dolaşılması gereken hedef listesi

Karınca ilgili düğüm ile haberleşerek kendisini göndereceği bir sonraki düğümü belirler. Kendi kopyasını (klonunu) aynı makinedeki veya farklı makinedeki hedef düğüme gönderir. Klon karınca hedef düğüme başarılı olarak ulaştıktan sonra kendisini sonlandırır.

### 5.4. Algoritma

Burada temel algoritma hareketli etmenin (karıncanın) çalışma algoritmasıdır. Düğümler olarak görülen statik etmenin temel görevi hareketli etmene bilgi sağlamaktır. Karınca düğümden aldığı feromon bilgileri ile bir sonraki adımda kendisini nereye yönlendirmesi gerektiğini seçer. Düğüme karıncanın hareketlerine göre kendi içerisindeki feromon bilgilerini günceller.

1. Eğer Seyahat Tamamlandı ise Düğüm gereklilik bilgileri ve Kendini Sonlandır()
2. Toplam Masrafı Hesapla()
3. Komsu Listesini AI (ACL ile)
4. Komsu Masraf Listesini AI (ACL ile)
5. Feromon Listesini AI (ACL ile)
6. Gidilecek Hedef Komşuyu KKO Algoritmasına Göre Seç()
7. Kendi Klonunu Çıkar()
8. Klon Karıncayı Hedef Komsuya Gönder()
9. Statik Etmene Feromon Güncellemesi için Bilgi Ver (ACL ile).
10. Kendini Sonlandır().

Şekil 4. Sistemin KKO Algoritması

Karıncanın çalışma algoritması Şekil 4'te gösterilmiş olup, karıncaların hareket etmek için yapmaları gereken olasılık hesaplaması Formül 1'e göre yapılmaktadır.

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [n_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [n_{il}]^\beta}$$

**Formül 1.** i şehrinde olan k karıncasının j şehrini seçme olasılığı

$p_{ij}^k$ ; i şehrinde olan k karıncasının j şehrini seçme olasılığı

$\tau_{ij}$ ; i ve j şehirleri arasındaki feromon miktarını gösterir.



$n_{ij}$  ; Sezgisel değeri ifade eder. ( $1/d_{ij}$ )

$\alpha$  ; Feromonu etkileyen feromon katsayısı

$\beta$  ; Sezgiselliği etkileyen sezgisellik katsayısı

$N_i^k$ ; k karıncası i şehrinde iken uğramadığı ve i şehirden sonra gidebileceği şehirler

### 5.5. Etmenler Arası Koordinasyon

Etmenler arası etkileşimde standardizasyonun sağlanması açısından FIPA-ACL etmen haberleşme dilinden faydalanılmıştır. Bu sayede, ihtiyaç duyulması durumunda FIPA spesifikasyonlarını destekleyen başka türden etmenlerle Statik ve hareketli etmenlerin haberleşme imkânı olmaktadır. Örnek bir 12 numaralı düğüme (statik etmene) sorulan bir feromon sorgu mesajı Şekil 5'te gösterilmiştir.

```
(request
:sender (:name Karınca17...)
:receiver (:name Dugum12...)
:ontology KKO
:language FIPA-SL
:protocolfipa-request
:content (feromon (Dugum (12) - (20)))
)
```

Şekil 5. Örnek bir FIPA-ACL Sorgusu

## 6. KONUYLA İLGİLİ YAPILMIŞ ÇALIŞMALAR

2009 yılında Wang ve arkadaşları tarafından yapılan çalışmada [22] çoklu etmen sistemlerin karınca koloni algoritmalarını kullanan seyyar satıcı problemlerini çözerken veri madenciliğinin kullanmasının yararlı olacağını öne sürmüştür. Veri madenciliği için gerekli bilgi seyyar satıcı kütüphanesi değerlendirme problemi eil51 üzerinden kullanılmıştır. Eil51 değerlendirme problemi 51 şehirden oluşmaktadır ve en iyi sonucu 426 rota uzunluğunda hesaplamıştır. Çözüm oluşturulurken karınca koloni optimizasyonlarında kullanılan seçim formülleri veri madenciliğinden elde edilen verilerde dikkate alınarak formüle edilmiştir.

Çok etmenli sistem üç katmanlı bir yapı üzerinde çalıştırılmıştır. Birinci katmanda karınca koloni optimizasyon etmeni, ikinci katmanda genetik algoritma etmeni ve üçüncü katmanda da hızlı yerel arama etmeni çalıştırılmıştır. Sonuçta öncelikle veri madenciliğinin seyyar satıcı problemleri üzerinde kullanılabildiği ve hibrit yapıdaki çoklu etmen sisteminin daha optimize sonuçlar verdiği gösterilmiştir.

2010 yılında Sorin Ilie ve Costin Badica tarafından yapılan çalışmada [21, 23] karınca koloni optimizasyonlarını kullanarak seyyar satıcı problemlerini çözebilen yeni bir yapı tasarlanmıştır. Bu yapıya “Dağıtık mimari üzerinde KKO” (ACODA (Ant Colony Optimization on a Distributed Architecture)) adı verilmiştir. Bu çalışmada oluşturulan yapının fonksiyonları, teknolojisi ve algoritması anlatılmış ve bu yapı üzerinde çalıştırılmış olan denemelerin sonuçları verilmiştir. JADE platformunda oluşturulan yapı üzerinde gerçekleştirilen denemeler dağıtık mimari üzerinde karınca koloni optimizasyonlarının uygulanabilirliğini ve ölçeklendirilebilirliğini desteklediğini açıkça göstermiştir.

2011 Ilie ve arkadaşları tarafından yapılan çalışmada [24] önceden geliştirilmiş olan “Dağıtık mimari üzerinde karınca koloni optimizasyonu” ile seyyar satıcı problemleri çözülmüştür. Bu çalışma dağıtık çoklu etmen arakatmanı üzerinde çalışan karınca koloni optimizasyon algoritmalarının daha iyi sonuç vermesi için bir bilişim mimarisi önermektedir. Çalışma sonucunda dağıtık bilişim mimarisi üzerinde çalışan karınca koloni optimizasyon algoritmalarının daha verimli ve ölçeklenebilir olduğu görülmüş ve yeni etmen tabanlı dağıtık karınca koloni optimizasyonlarının anlaşılmasını kolaylaştırmıştır. Genelde seyyar satıcı problemleri üzerinde yapılan çalışmalar sıralı çözümü ele alırken bu çalışmada karıncanın gideceği yollar alt yollara bölünmüş bu yolların her birine de farklı karıncalar görevlendirilmiştir. Oluşturulan algoritma eil51, st70, kroA100, ch150, gr666 seyyar

satıcı kütüphanesi değerlendirme problemleri üzerinde çalıştırılmış ve elde edilen değerler tablolarla gösterilmiştir. Sonuçta parçalı yollar ve karıncalar üzerinde daha detaylı çalışılması gerektiği ve ileride sonuçların daha da iyileştirilebileceği gösterilmiştir.

2007 yılında Chi-Bin Chenga ve Chun-Pin Mao tarafından yapılan çalışmada [25] bazı özellikleri değiştirilmiş bir karınca koloni optimizasyonu seyyar satıcı problemlerini zaman kısıtlı olarak çözülmüştür. Zaman kısıtlı seyyar satıcı problemleri karıncanın kendisine ayrılan zaman içerisinde en düşük uzaklık maliyetiyle bütün şehirleri ziyaret ederek dolaşmasını içerir. Geliştirilen bu algoritmada problemi zaman kısıtlarına ayırabilmek için iki tane yerel gömülü sezgisel metot kullanılmıştır. Seçilmiş belli bir grup değerlendirme problemi üzerinde elde edilen sayısal sonuçlar karınca koloni sistemi zaman kısıtlı seyyar satıcı problemlerinin seyyar satıcı zaman kısıtlı problemlerini çözmek için varolan ACS-Time [26] ile karşılaştırılabilir olduğunu göstermiştir.

## 7. SONUÇLAR VE GELECEK ÇALIŞMALAR

Bu çalışmada KKO optimizasyonu algoritmasının dağıtılmış ortamda bir çoklu etmen sistemi içerisinde uygulanması incelenmiştir. Hareketli etmenlerin kullanılması ile sistemin ölçeklenebilirliği ve paralelleştirilebilirliği artırılmış ve karıncaların bir otonom etmen yapısında geliştirilmesi ile gerçek dünyadaki evrimsel çözüm yaklaşımı sanal ortamda daha doğru şekilde gerçekleştirilmiştir. Bu sayede sistemin hesaplama yükü farklı makineler üzerine dağıtılmış ve merkezi KKO sisteminden daha gürbüz ve hata hoşgörülü bir sistem geliştirme imkânı sağlanmıştır. Karınca modelinin etmen yapısına eklenebilecek ilave haberleşme ve koordinasyon becerileri ile de sistem performansının daha da artırılabilmesi değerlendirilmektedir.

Gezgin satıcı problemi Yöneylem Araştırması alanında farklı çözüm yöntemlerin denenmesinde kullanılan çok genel bir problemdir. Özellikle evrimsel algoritmaların uygulanmasına da olanak sağlayan bu problem alanında etmen tabanlı KKO Algoritmasının kullanılabilmesini değerlendirmekteyiz. Bu kapsamda önerdiğimiz sistemin yapısı, kullanılan etmen yapıları, algoritmaları ile birlikte bu bildiride özetlenmiştir. İleriki çalışmalarda bu sistemin test sonuçları ve ölçeklenebilir yapısının sunulması amaçlanmaktadır.

## 8. KAYNAKLAR

- [1] Dorigo, M., Blum, C., “Ant Colony Optimization Theory : A Survey”, 2005.
- [2] M. Pedemonte, S. Nesmachnow, H. Cancela, A survey on parallel ant colony optimization, Applied Soft Computing (in press), Corrected Proof, 2011.
- [3] Dorigo, M., “Optimization, Learning and Natural Algorithms, PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
- [4] Blum C., “Ant colony optimization: Introduction and recent trends”, Physics of Life Reviews 2, 2005, 353-373
- [5] Bonabeau, E., Dorigo, M., Theraulaz, G., “Swarm Intelligence : From Natural to Artificial, Oxford University Press”, 1999.
- [6] Middendorf, M., Reischle, F., Schmeck, H., “Information Exchange in Multi Colony Ant Algorithms”, Volume: 1800, Publisher: ACM Special Interest Group on Computer Architecture (SIGARCH), and IEEE Computer Society, 645-652
- [7] Özdemir Y.S., “Karınca Kolonisi Algoritması İle Bilgisayar Ağlarının Topolojik En İyilenmesi”, Başkent Üniversitesi, 2008.
- [8] Dorigo, M., Maniezzo, V., Colomi A., ”The Ant System : Optimization by a colony

- of cooperating agents”, IEEE Transactions on Systems, Man, and Cybernetics, 1996, 29-41
- [9] Dorigo, M., Maniezzo, V., Colomi A., “An Investigation of Some Properties of an Ant Algorithm”, Parallel Problem Solving from Nature - PPSN II: International Conference on Evolutionary Computation - The Second Conference on Parallel Problem Solving from Nature, pages 509-520. North-Holland, Amsterdam, The Netherlands, 1992.
- [10] Gambardella L.M, Dorigo M., “Ant-Q : A Reinforcement Learning Approach to the Traveling Salesman Problem”, Twelfth International Conference on Machine Learning, 1995, 252-260.
- [11] Dorigo M., Gambardella L.M., “A Study of some Properties of Ant-Q”, Proceedings of PPSN96 International Conference on Parallel Problem Solving from Nature, H.-M. Voigt, Springer-Verlag, Berlin, 1996, 656-665.
- [12] Dorigo, M., Gambardella, L. M., “Ant Colony System: A Cooperative Learning Approach to The Travelling Salesman Problem”, IEEE Transactions on Evolutionary Computation, 53-66, 1997.
- [13] Stützle T, Hoos HH., “MAX-MIN Ant system.”, Future Generat Comput Syst, 2000, 889-914.
- [14] Bullnheimer B., Hartl R., Strauss C., “A new rank-based version of the Ant System: A computational study.”, Central European J Operations Res/Econom 1999;7(1):25-38
- [15] Blum C., Dorigo M., “The hyper-cube framework for ant colony optimization.”, IEEE Trans Syst Man Cybernet Part B 2004;34(2):1161-72.
- [16] Engelbrecht, A. P., “Fundamentals of Computational Swarm Intelligence”, John Wiley & Sons Ltd., England, 2005.
- [17] Luck, M., McBurney, P., Shehory, O., Willmott, S., Agent Technology: Computing as Interaction : A Roadmap for Agent-Based Computing, AgentLink III, <http://www.agentlink.org/roadmap/al3rm.ppf>, 2005.
- [18] Russell, S., Norvig, P., "Artificial Intelligence : A Modern Approach" (2nd Edition).
- [19] Wooldridge M., Jennings N., “Intelligent Agents : Theory and Practice.”, Manchester Metropolitan University, Department of Computing, UK, 1995.
- [20] <http://www.fipa.org/>
- [21] Ilie, S., Badica, C., “Effectiveness of Solving Traveling Salesman Problem Using Ant Colony Optimization on Distributed Multi-Agent Middleware”, IEEE, syf 197-203, 2010.
- [22] Wang, S.Q., Xu Z.Y., “Ant Colony Algorithm Approach for Solving Traveling Salesman with Multi-agent” WASE International Conference on Information Engineering, 2009.
- [23] Ilie, S., Badica, A., Badica, C., “Distributed agent-based ant colony optimization for solving traveling salesman problem on a partitioned map”. WIMS '11 Proceedings of the International Conference on Web Intelligence, Mining and Semantics Symposium of Intelligent Distributed Computing, Article No. 23, 2011.
- [24] Ilie, S., Badica, C., “Distributed multi-agent system for solving traveling salesman problem using ant colony optimization”. In Proc. 4th International Symposium of Intelligent Distributed Computing, IDC'2010, volume 315, pages 119-129. Springer-Verlag, 2010.
- [25] Chenga, C., Maob, C., “A modified ant colony system for solving the travelling salesman problem with time Windows”, Mathematical and Computer Modelling 46, 2007, 1225-1235.
- [26] Gambardella, L.M., Taillard, E.D., Agazzi, G., “A multiple ant colony system for vehicle routing problems with time Windows”, New Ideas in Optimization, McGrawHill, London, UK, 1999, 63-76.