

Türkiye Haritası Üzerinde Gezgin Satıcı Probleminin Karınca Sistemleri ile Çözülmesi

Emrah Yavuz, Ender Özcan, and Emin Erkan Korkmaz

Bilgisayar Mühendisliği Bölümü, Yeditepe Üniversitesi, İstanbul, Türkiye
{eyavuz, eoacan, ekorkmaz}@cse.yeditepe.edu.tr

Öz. Gezgin satıcı problemi birleşimsel bir optimizasyon problemidir. Karıncaların yol bulmak için kullandıkları yöntemden esinlenilerek oluşturulan karınca sistemleri ise birçok optimizasyon probleminin çözümünde başarıyla kullanılmışlardır. Bu makalede gezgin satıcı problemi için değişik araştırmacılar tarafından kullanılan farklı karınca sistemleri toplu olarak yine bu alanda kullanılan problem setleri üzerinde analiz edilmiştir. Problem örnekleri arasında Türkiye haritası da dahil edilmiştir. Deney sonuçları maksimum-minimum karınca sisteminin örnek problemler üzerinde en iyi performansa sahip olduğunu göstermektedir.

1 Giriş

Gezgin satıcı problemi ilk olarak K. Menger[10] tarafından yaklaşık seksen yıl önce tanımlanmış zor bir optimizasyon problemidir. Bu problemde bir satıcının birbirinden farklı n adet şehri ziyaret etmesi gerekmektedir. Satıcının turu başladığı şehirde sona ermeli, her şehir sadece bir kere ziyaret edilmelidir. Problem bu koşulları sağlayan olası turlar içinde minimum uzunluğa sahip olan turu bulmaktır. Bu problemi çözmek için araştırmacılar tarafından birçok değişik yaklaşım ve algoritma kullanılmıştır. Halen daha konu üzerine araştırmacıların ilgisi devam etmektedir [17, 12, 13, 6–8].

Karınca sistemleri ise oldukça yeni ama birçok optimizasyon problemi üzerinde başarı ile uygulanmış algoritmalar. İlk olarak A. Coloni [3] tarafından ortaya konulan bu yaklaşım karınca kolonilerinin barınakları ve yiyecek kaynağı arasındaki en kısa yolu bulmak için kullandıkları yöntemden esinlenerek oluşturulmuştur. Karıncalar bu en kısa yolu fermon adı verilen bir kimyasal madde kullanarak bulurlar. Bir yolda ilerleyen her karınca belirli aralıklarla bu kimyasal yola bırakmaktadır, arkadan gelen karıncalar ise karşısına bir yol ayrımı geldiğinde bu maddenin daha çok yer aldığı seçeneği takip ederler. Bir karınca sürüsü yeni bulunan bir yiyecek kaynağından barınaklarına yiyecek taşımaya başladığında, ilk başta karıncalar rastgele hareket edeceklerdir. Ancak, bir süre sonra en kısa yoldan barınağa ulaşan karıncaların, yiyecek kaynağına daha sık gidip gelebilmeleri nedeniyle onların izlediği yoldaki fermon miktarı artacak ve diğer karıncalar da bu fazlalaşan kimyasal takip ederek en kısa yol üzerinden yiyecek taşımaya başlayacaklardır.

Kısaca karınca sistemi olarak adlandırılan bu yöntem gezgin satıcı problemine de daha önce uygulanmıştır [5, 14]. Bu çalışmada gezgin satıcı problemi için araştırmacılar tarafından kullanılan farklı karınca sistemlerinin analizi amaçlanmıştır. Bu analiz yine alanda kullanılan problem setleri üzerinde gerçekleştirilmiştir. Ancak daha önceki problem örnekleri arasına Türkiye haritası da dahil edilmiş ve bu yeni örnek üzerinde de var olan yaklaşımlar denenmiştir. Türkiye haritası üzerinde elde edilen sonuçlar daha önce genetik algoritma yöntemi ile bulunan sonuçlarla [11] aynıdır. Diğer örnek problemler üzerinde edilen sonuçlar da daha önceki çalışmalarla uyum içerisindedir.

2 Karınca Sistemleri ve Gezgin Satıcı Problemi

Gezgin satıcı problemi uzunca bir süredir üzerinde yaygın bir şekilde araştırma yürütülmüş bir problemidir. Problemin zorluğu onu yeni algoritma ve yöntemlerin sınındığı bir alan haline getirmiştir. Bu problem üzerinde gösterilen iyi bir performans, öne sürülen yeni yaklaşımın etkinliğinin bir kanıtı olarak algılanmaktadır.

Problem her bir bağlantısının bir ağırlığa sahip olduğu, bütün bir C çizgesi ile temsil edilebilir, $C = (D, B)$. Burada D çizgedeki düğümlerin kümesini, B ise tüm düğümleri birbirine bağlayan bağlantıların kümesini temsil etmektedir. Her bir bağlantı için belirli bir $a_{i,j}$ ağırlığı tanımlanmıştır ve bu ağırlık i ve j düğümleri arasındaki bağlantının uzunluğunu temsil etmektedir. Problem böyle bir çizgede minimum uzunluktaki Hamiltonian devresini bulabilmektir. Simetrik gezgin satıcı probleminde şehirler arasındaki uzaklık ya da çizgedeki bağlantıların ağırlığı bir düğümden diğerine geçiş yönünden bağımsızdır. Yani bütün bağlantılar için $d_{i,j} = d_{j,i}$. Asimetrik gezgin satıcı probleminde ise bu özelliği sağlamayan en az bir adet bağlantı bulunmalıdır. Bu çalışmada simetrik gezgin satıcı problemi ele alınmıştır. n adet şehirden oluşan bir problem örneğinde $(n - 1)!$ adet değişik tur var olduğu göz önüne alınırsa problemin neden klasik arama algoritmaları ile çözülmediği açıktır. Evrimsel algoritmalar, karınca sistemleri gibi daha etkin optimizasyon yaklaşımları bu yüzden bu alanda ön plana çıkmaktadır.

Gerçek karıncalar yollarını bulmak için herhangi bir görsel veri kullanamazlar. Tek bir karıncanın kabiliyetleri oldukça sınırlı olsa bile, bir koloni olarak karıncalar yiyecek ve barınakları arasındaki en kısa yolu bulmak konusunda oldukça başarılıdırlar. Bir karınca grubuna bu bütünsel yeteneği sağlayan şey ise her bir karıncanın dolaşırken etrafa bıraktığı fermon adlı kimyasaldır.

Bu kimyasalın salgılanmasının nasıl bir optimizasyon yöntemine dönüştüğünü anlamak için figür 2'de verilen örnek faydalı olacaktır. Bu örnekte belirli bir yiyecek kaynağı ve barınakları arasında yol alan karıncaların önüne daha önceden bulunmayan bir engel çıkmıştır. Karınca kolonisinin çözmesi gereken problem bu engelin sağından mı yoksa solundan mı dolaşıldığında yeni durumdaki en kısa yolun oluşacağıdır. Engele ilk ulaşan karıncalar açısından problemin çözümüne ilişkin elde hiçbir veri bulunmamaktadır. Bu yüzden bu ilk karıncalar tamamen rastgele davranacaklar ve bir kısmı engelin sağından dolaşmaya çalışırken bir kısmı diğer yanından engeli aşmaya çalışacaktır. Bu iki karınca grubunun engelin

iki tarafına da fermon kimyasalını bırakmaları geriden gelen karıncalar açısından ilk başlarda yardımcı bir unsur olmayacaktır. Ancak şans eseri daha kısa yolu tercih eden karıncalar barınaklarına daha hızlı ulaşacaklar ve yiyeceklerini bırakıp tekrar engelin bulunduğu yere dönmeleri diğer yolu seçen karıncalara göre daha hızlı olacaktır. Bu da engelin doğru tarafında bulunan fermon maddesinin artmasına sebep olacak ve yavaş yavaş diğer karıncaların da bu maddenin daha fazla olduğu yöne yönelmeleri sebebiyle bir süre sonra tüm karıncalar bir koloni olarak keşfedebildikleri bu en kısa yolu kullanmaya başlayacaklardır.

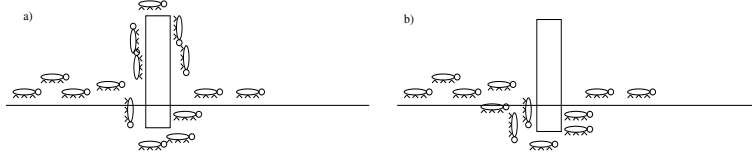


Fig. 1. Şeklin ilk bölümünde görüldüğü gibi karıncalar yeni bir engelle karşılaştıklarında engelin iki yanından da dolaşacaklar, ama ikinci bölümde gösterildiği gibi bir süre sonra hepsi kısa olan yolu tercih edeceklerdir.

Karıncı optimizasyon sistemleri, karıncaların yukarıda sözü edilen yeteneklerinden esinlenerek tasarlanmıştır. Bu sistemlerde her bir karınca bir yazılım ajanı olarak kurgulanmıştır. Her yapay karınca ajanının davranışı o anki fermon değerlerine göre lokal ve türel bir tercihe bağlıdır. Karıncaların seçtiği davranışlara göre de algoritmanın belirli aşamalarında varolan fermon değerleri güncellenmektedir. Davranış seçimi ve fermon değerlerinin güncellenmesi aşamalarında değişik tercihler varolabilir. Bu konuda birbirinden farklı yaklaşımlar geliştirilmiş ve her yaklaşım farklı bir karınca sistemi olarak adlandırılmıştır. Bu çalışmada yapılan deneylerde *Temel karınca sistemi*, *Seçkinci karınca sistemi*, *Karıncı koloni sistemi*, *Maksimum-minimum karınca sistemi*, *En iyi-en kötü karınca sistemi* ve *Dereceye göre karınca sistemi* kullanılmıştır.

Gezgin satıcı problemi üzerinde karınca sistemi yöntemi uygulanmasında rastgele şehirlere yerleştirilen m adet karınca, döngüsel olarak daha ziyaret edilmiş şehirleri belirli bir stokastik modele göre ziyaret ederek uygun bir turu oluşturmaya çalışacaktır. Karıncalar gidecekleri şehirleri seçerken şehirler arasındaki bağlantılara ait fermon değerlerine göre karar vereceklerdir. Tabiki bir bağlantıdan daha çok karıncanın geçmesi o bağlantının fermon değerini belirli bir modele göre yükseltecektir. Her bir karıncanın tam bir tur oluşturması beklendiğinden, her karınca için o ana kadar o karıncanın ziyaret ettiği şehirlerin listesinin de algoritma içinde tutulması gerekmektedir.

2.1 Kullanılan Karınca Sistemlerinin Tanımı

Temel Karınca Sistemi: Deneyler için ilk kullanılan karınca sistemi *temel karınca sistemidir*. Bu sistemde i şehrinde bulunan bir karıncanın j şehrine geçmesinin olasılığı şöyle tanımlanmaktadır.

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{i \in \mathbf{N}_i^k} [\tau_{it}(t)]^\alpha \cdot [\eta_{it}]^\beta} \quad (1)$$

Bu eşitlikte, $\tau_{ij}(t)$, i ve j şehirleri arasındaki bağlantının sahip olduğu fermon miktarını belirlemektedir. \mathbf{N}_i^k ise, k karıncası tarafından henüz ziyaret edilmemiş şehirlerin kümesidir ve j şehri de bu kümenin bir elamanı olmalıdır. d_{ij} , i ve j şehirleri arasındaki uzaklık ise, $\eta_{ij} = \frac{1}{d_{ij}}$ eşitliği ile bu değer şehir seçiminde buluşsal (*heuristic*) bir bilgi olarak kullanılması sağlanmıştır. α ve β parametreleri ise buluşsal bilgi ve fermon miktarlarının karar aşamasını ne kadar etkileyeceğini belirleyen parametrelerdir. Örneğin α değeri sıfır yapıldığında, sürekli en yakındaki şehir seçileceği için algoritma stokastik saldırgan (*greedy*) bir algoritmaya dönüşecektir.

Her bir karınca kendi turunu oluşturduğunda fermon değerleri güncellenmektedir. Bu işlem başta tüm bağlantılardaki fermon değerini belirli bir miktar düşürdüktan sonra, her karıncanın kullandığı bağlantıların fermon değerini yükseltilmesi şeklinde gerçekleştirilmektedir. Aşağıdaki eşitlik bu güncellemeyi tanımlamaktadır.

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (2)$$

Bu eşitlikte ρ parametresi ($0 \leq \rho \leq 1$) her aşamada ne kadar fermon miktarının bağlantılardan silineceğini belirtmektedir. $\Delta \tau_{ij}^k(t)$ ise k karıncasının (i, j) bağlantısına eklediği fermon miktarıdır. Bu değer

$$\Delta \tau_{ij}^k(t) = \frac{1}{L^k(t)} \quad (3)$$

olarak tanımlanmıştır. Bu eşitlikte $L^k(t)$ değeri k karıncasının oluşturduğu turun uzunluğudur. Genel olarak bir bağlantının birden fazla karınca tarafından ziyaret edileceği düşünüldüğünde, daha kısa turlarda yer alan bağlantıların fermon miktarının daha çok artacağı ve ilerki aşamalarda daha çok tercih edilecekleri açık olacaktır [16].

Seçkin Karınca Sistemi: Temel karınca sistemi üzerine yapılan ilk iyileştirme seçkin bir yaklaşım geliştirmek olmuştur. Bu yaklaşımda önceki sistemden farklı olarak o ana kadar bulunan en iyi tura ait bağlantıların fermon değerleri bir önceki sisteme göre daha güçlü bir biçimde artırılmaktadır. Bunun için bu tür bağlantılara ekstra bir fermon değeri eklenmektedir.[1]

Karınca Koloni Sistemi: Karınca koloni sistemi de temel karınca sistemini geliştirmek için öne sürülen bir yaklaşımdır [9]. Bu yaklaşım temel sisteme göre üç noktada farklıdır. Öncelikle bu yaklaşımda karıncalar gidilecek şehir seçiminde belirli bir olasılıkla o ana kadar oluşan fermon izlerinin gösterdiği en iyi bağlantıyı tercih etmektedirler. Bu olasılık gerçekleşmediği takdirde temel karınca sisteminin yöntemine dönülmektedir. Bir karınca bir bağlantıyı kullandığında ise o bağlantıdaki fermon miktarı azaltılmaktadır. Ayrıca temel sis-

temde tüm karıncaların hareketleri fermon miktarlarını değiştirirse de, bu yaklaşımda sadece en iyi karıncanın bu güncellemeyi yapmasına izin verilmektedir.

Maksimum-Minimum Karınca Sistemi: Bu sistemde şehir seçimi temel karınca sisteminde olduğu gibi yapılmakta ancak koloni sisteminde olduğu gibi fermon değerlerinin güncellenmesi sadece en iyi karıncaya bırakılmaktadır. Ayrıca arama işleminin durağanlaşması ihtimaline önlem olarak varolabilecek fermon değerleri için alt ve üst limitler belirlenmektedir. Arama işleminin başlangıcında fermon değerleri tüm bağlantılar için en üst sınır değeri olarak verilmekte ve böylece arama sürecinin başlangıç kısımlarında daha genel bir araştırma yapılması sağlanmaktadır [15].

En İyi - En Kötü Karınca Sistemi: Bu yaklaşımda temel karınca sistemi bir takım evrimsel algoritma kavramları kullanılarak daha da geliştirilmeye çalışılmıştır. Örneğin varolan fermon değerleri bir mutasyon operatörü ile rastgele değiştirilerek arama sürecinin çeşitliliğinin sürdürülmesi amaçlanmıştır. Yine evrimsel algoritmalarda olduğu gibi arama işleminde bir sıkışma tespit edildiğinde arama süreci tekrar baştan başlatılmaktadır. Ayrıca en iyi karıncanın fermon değerlerine katkısı fazlalştırıldığı gibi en kötü performansı gösteren karıncanın ziyaret ettiği bağlantıların fermon değerleri de düşürülmektedir [4].

Dereceye Göre Karınca Sistemi: Yine temel karınca sisteminin daha geliştirilmiş hali olan bu sistemde bir grup iyi karıncanın fermon değerlerini güncellemesine izin verilmektedir. Bu grubun elemanları bulabildikleri tur uzunluklarına göre sıralanmakta, ve bu sıralamada elde ettikleri derece ölçüsünde fermon değerlerini değiştirme gücüne sahip olmaktadır [2].

3 Deneysel Sonuçlar

Var olan karınca sistemlerinin performansının artırılması için kullanılan bir başka yöntem de karınca sistemi algoritmasının her adımından sonra (karıncaların her yeni bir turu oluşturmasından sonra) basit bir yerel arama gerçekleştirerek bulunan turların biraz daha iyiye götürülmesidir [16]. Yaygın bir biçimde kullanılan $2 - opt$ ve $3 - opt$ yerel arama methodları bu çalışmada da kullanılmıştır. Bu yerel arama algoritmaları var olan bir turdaki iki ($2 - opt$) ya da üç ($3 - opt$) bağlantının değiştirilerek daha iyi bir tur elde edilip edilmeyeceğini kontrol etmektedirler. Kontrol edilen bağlantı sayısı az tutulduğu için bu arama işleminin, genel algoritmanın çalışma zamanına çok büyük yükü olmamasına rağmen, çözüme ulaşma sürecini oldukça hızlandırdığı gözlemlenmektedir [16].

Bir önceki bölümde açıklanan altı ayrı karınca sistemi bir dizi deneyle analiz edilmiştir. Bu sistemler bazı farklılıklara sahip olsalar da, tüm sistemlerde kullanılması gereken bir takım deneysel parametreler bulunmaktadır. Bu farklı sistemlerin birbirleri ile performans açısından karşılaştırılabilmesi için tüm deneylerde bu ortak parametreler için aynı değerler kullanılmıştır. Kullanılan bu değerler, deneme sayısı= 50, karınca sayısı= 25, $\alpha = 1$, $\beta = 2$ ve $\rho = 0.5$ olarak belirlenmiştir.

Deney aşamasında ilk olarak arama işlemini yönlendirmede oldukça önemli olan *Beta* parametresi üzerinde durulmuş ve bu parametrenin karınca sistem-

Tablo 1. Kullanılan örnek problemler ve özellikleri. A:Problem, B:Şehir sayısı, C: Bilinen en iyi çözüm. *tr.tsp* ismiyle verilen Türkiye problemi orta zorlukta bir problem olarak düşünülebilir.

A	eil.tsp	tr.tsp	kroA.tsp	d1.tsp	lin.tsp	pcb.tsp	att.tsp	rat.tsp	pcb.tsp	d.tsp	pr2392.tsp
B	51	81	100	198	318	442	532	783	1173	1291	2329
C	426	76242	21282	15780	42029	50778	27686	8806	56892	50801	378032

Tablo 2. Beta parametresinin etkisi. Tabloda yer alan rakamlar her beta değeri için örnek problemlerin kaçında optimum çözümün bulunduğunu belirtmektedir.

	Temel Sis.	Koloni Sis.	Maks-Min	En iyi-En kötü	Seçkinci	Dereceli
$\beta = 1$	4	6	6	5	6	7
$\beta = 2$	11	10	11	10	11	9

leri üzerindeki etkisi ölçülmeye çalışılmıştır. Tablo 2’de de görüldüğü üzere bu parametrenin iki değerine sahip olduğunda karınca sistemlerinin daha başarılı olduğu gözlemlenmiş ve bu aşamadan sonraki deneyler için *Beta* parametresi iki olarak sabitlenmiştir.

Tablo 3. Karınca sistemlerinin başarı sırası. Aynı değere ulaşan yöntemler aynı dereceyle gösterilmişlerdir.

	Temel Sis.	Koloni Sis.	Maks-Min	En iyi-En kötü	Seçkinci	Dereceli
eil51.tsp	1	1	1	1	1	1
T81.tsp	1	1	1	1	1	1
kroA100.tsp	1	1	1	1	1	1
d198.tsp	1	1	1	1	1	1
lin318.tsp	2	1	1	1	1	1
pcb442.tsp	4	1	1	2	1	3
att532.tsp	5	4	1	1	2	3
rat783.tsp	5	3	1	2	1	4
pcb1173.tsp	6	4	1	2	3	5
d1291.tsp	5	3	4	1	2	2
pr2392.tsp	6	3	1	4	2	5
Ortalama	3,36	2,09	1,27	1,55	1,45	2,45

Tablo 3 ise karınca sistemleri arasında yapılan performans karşılaştırması sonucu oluşturulmuştur. Deney sonuçları lokal aramanın etkili olduğunu ve tüm sistemler için 3-opt aramasının en iyi sonuçları verdiğini göstermiştir. Daha sonra 3-opt araması kullanılan durumda bulunan en iyi turlara göre karınca sistemleri derecelendirilmişlerdir. Tabloda 1 en iyi dereceyi, 6 ise en kötü dereceyi göstermektedir. Aynı sonuçlara ulaşan sistemlere aynı dereceler verilmiştir. Tabloda görüldüğü gibi göreceli olarak kolay olan problem setlerinde tüm sistemler aynı performansı gösterirken, problem örnekleri zorlaştıkça maksimum-

minimum karınca sistemi başarısı ön plana çıkmıştır. Tablonun son satırında yer alan ortalama dereceler de bu sistemin diğer yaklaşımlara göre daha iyi bir performansa sahip olduğunu göstermektedir.

Tablo 4. Türkiye haritası üzerindeki sonuçlar.

	Temel Sis.	Koloni Sis.	Maks-Min	En iyi-En kötü	Seçkinci	Dereceli
Ort. (iterasyon)	47,3	6,3	10,6	21	5,7	6,4
Std. sapma	44,06	2	6,26	38,12	2,83	4,2
Ort. (en iyi tur)	76242	76242	76242	76242	76253,4	76257,2
Std. sapma	0	0	0	0	18,36	19,62
Ort. (zaman-sn)	0,19	0,03	0,04	0,08	0,02	0,2
Std. sapma	0,18	0,01	0,02	0,15	0,01	0,2

Türkiye haritası üzerinde elde edilen sonuçlar ise tablo 4’de görülebilir. Bu problem görece kolay problemlerden olduğu için sistemlerin performansları arasında çok büyük farklılıklar gözlemlenmemiştir.

Bulunan en iyi Türkiye turu da aşağıda verilmiştir. Türkiye haritası üzerinde daha önce genetik algoritma yöntemi ile en kısa tur hesaplanmıştır [11]. Karınca sistemi ile elde edilen tur, daha önceki çalışmada bulunan tur ile aynıdır.

Türkiye haritası üzerinde bulunan en kısa tur: HAKKARİ, ŞIRNAK, ŞİRT, BİTLİS, MUŞ, BİNGÖL, ERZİNCAN, TUNCELİ, ELAZIĞ, DİYARBAKIR, BATMAN, MARDİN, ŞANLIURFA, ADIYAMAN, MALATYA, KAHRAMANMARAŞ, GAZİANTEP, KİLİS, HATAY (Antakya), OSMANIYE, ADANA, KAYSERİ, YOZGAT, NEVŞEHİR, NİĞDE, İÇEL (Mersin), KARAMAN, ANTALYA, BURDUR, AFYON, ISPARTA, KONYA, AKSARAY, KIRŞEHİR, KIRIKKALE, KARABÜK, BARTIN, ZONGULDAK, BOLU, DÜZCE, SAKARYA (Adapazarı), BİLECİK, KÜTAHYA, UŞAK, DENİZLİ, MUĞLA, AYDIN, İZMİR, MANİSA, BALIKESİR, ÇANAKKALE, EDİRNE, KIRKLARELİ, TEKİRDAĞ, İSTANBUL, BURSA, YALOVA, KOCAELİ (İzmit), ESKİŞEHİR, ANKARA, ÇANKIRI, KASTAMONU, ÇORUM, SİNOP, AMASYA, SAMSUN, TOKAT, SİVAS, ORDU, GİRESUN, GÜMÜŞHANE, TRABZON, BAYBURT, ERZURUM, RİZE, ARTVİN, ARDAHAN, KARS, AĞRI, IĞDIR, VAN

4 Özet ve Sonuç

Bu çalışmada karınca kolonilerinin davranışlarından esinlenerek oluşturulan değişik karınca sistemleri gezgin satıcı problemi üzerinde incelenmiştir. Bu değişik yaklaşımlar bir araya toplanıp, aynı deneysel kurulum kullanılarak sistemlerin birbirleri ile performans açısından karşılaştırılması sağlanmıştır. Daha önceden kullanılan problem setlerine Türkiye haritası da dahil edilmiş ve Türkiye il-lerinin tümünü dolaşan en kısa tur da deney sonuçlarında sunulmuştur. Bundan sonra planlanan çalışma karınca sistemlerinin performansını yükseltmek için kullanılan lokal arama üzerine yoğunlaşacaktır. Amaçlanan bu lokal arama yerine daha etkin bir arama yöntemi kullanarak karınca sistemlerini hibrid bir yapıya kavuşturmak ve daha da geliştirmektir.

References

1. K. D. Boese, A. B. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16:101–113, 1994.
2. Bernd Bullnheimer, Christine Strauss, and Richard F. Hartl. A new rank based version of the ant system - A computational study, April 25 1997.
3. Alberto Colomi, Marco Dorigo, and Vittorio Maniezzo. Distributed optimization by ant colonies. In *Proceedings of ECAL91 - European Conference on Artificial Life*,. Elsevier Publishing, 1991.
4. Oscar Cerdón, Iñaki Fernández de Viana, and Francisco Herrera. Analysis of the best-worst ant system and its variants on the QAP. *Lecture Notes in Computer Science*, 2463:228–??, 2002.
5. M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–56, 1997.
6. Gutin, Yeo, and Zverovich. Traveling salesman should not be greedy: Domination analysis of greedy-type heuristics for the TSP. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 117, 2002.
7. Håkan Jonsson. The Traveling Salesman Problem for lines in the plane. *Information Processing Letters*, 82(3):137–142, May 2002.
8. Mr Jozef Kratica. One improvement to "nearest neighbor" method for solving "traveling salesman" problem, March 21 2001.
9. Dorigo M. and L.M. Gambardella. A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–66, 1997.
10. Karl Menger. Das botenproblem. *Ergebnisse eines Mathematischen Kolloquiums*, 1932.
11. E. Ozcan and M. Erenturk. A brief review of memetic algorithms for solving euclidean 2d traveling salesrep problem. In *13th Turkish Symposium on Artificial Intelligence and Neural Networks*, pages 99–108, 2004.
12. Luis Paquete and Thomas Stützle. A Two-Phase Local Search for the Biobjective Traveling Salesman Problem. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 479–493, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.
13. A. Plebe and A. M. Anile. A neural-network-based approach to the double traveling salesman problem. *Neural Computation*, 14(2):437–471, February 2002.
14. T. Stützle, A. Grün, S. Linke, and M. Rüttger. A comparison of nature inspired heuristics on the traveling salesman problem. In Hans-Paul Schwefel Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, editor, *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, Paris, France, September 16-20 2000. Springer Verlag. LNCS 1917.
15. Thomas Stützle and Holger H. Hoos. MAX-MIN Ant System. *Future Generation Computer Systems*, 16(8):889–914, June 2000.
16. Stutzle T. and M. Dorigo. Aco algorithms for the traveling salesman problem. *Evolutionary Algorithms in Engineering and Computer Science*, 1999.
17. Christopher White and Gary Yen. A hybrid evolutionary algorithm for traveling salesman problem. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 1473–1478, Portland, Oregon, 20-23 June 2004. IEEE Press.