

KARELİ ATAMA PROBLEMİ İÇİN ANTSIMULATED ALGORİTMASI

Nihan Çetin Demirel

Yıldız Teknik Üniversitesi, Endüstri Mühendisliği Bölümü, İstanbul

Duran Toksarı

Yıldız Teknik Üniversitesi, Endüstri Mühendisliği Bölümü, İstanbul

Özet: Karınca Algoritması optimizasyon problemlerinin çözümü için gerçek karınca kolonilerinin davranışlarından esinlenilerek geliştirilmiştir. Bu bildiri Kareli Atama Probleminin (Quadratic Assignment Problem-QAP) çözümü için lokal arama prosesi Tavlama Benzetimi (Simulated Annealing) olan Karınca Algoritmasını sunmuştur. Arama alanının araştırılmasında karıncalar tarafından yayılan fenomen kokusu seviyesinin değişimi kullanılmıştır. Bu çalışmada C++ programlama dili ile kodlanan AntSimulated Algoritması literatürdeki mevcut problemler kullanılarak analiz edilmiştir ve mevcut problemler için denenmiş sezgisellerin bazılarıyla karşılaştırılmıştır.

Anahtar Kelimeler: Kareli Atama Problemleri, Karınca Algoritması, Tavlama Benzetimi, Meta-Sezgiseller

Abstract: Ant Algorithm is developed from the inspiration of the behaviors of real ant colonies to solve optimization problems. This paper presents the ant colony to solve the quadratic assignment problem (QAP) of which local search process is Simulated Annealing. In the exploration of the search space, the evolution of pheromones which are laid on the ground by ants is used. In this work, AntSimulated Algorithm that is coded by C++ programming language is analysed by using current problems in the literature and is compared with other some heuristics.

Keywords: Quadratic Assignment Problem, Ant Algorithm, Simulated Annealing, Metaheuristics

1. Giriş

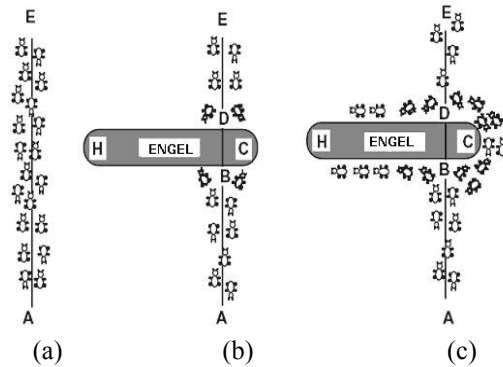
Karınca Algoritması (KA), genetik algoritmalar, tavlama benzetimi (TB), yapay sinir ağları gibi doğal sistemlerden esinlenerek geliştirilmiş kombinatoriyal problemlerde kullanılan bir algoritmadır. Dorigo tarafından geliştirilen KA'nın temel düşüncesi, yemlerine ulaşmada karınca kolonisinin davranışından esinlenerek geliştirilmiştir (Dorigo, 1992). KA gezgin satıcı problemi (Dorigo v.d., 1996), atölye tipi iş çizelgeleme (Colomi v.d., 1994), araç rotalama (Bullheimer v.d., 1997) v.b. birçok farklı optimizasyon problemleri için başarılı sonuçlar vermiştir.

Bu çalışmada öncelikle karınca algoritmasının kombinatoriyal optimizasyonda kullanımı hakkında temel bilgi verilecektir. Daha sonraki kısımda, kareli atama problemlerine (KAP) çözüm bulun, yerel araması TB ile yapılan *AntSimulated* algoritması incelenecektir. Son olarak literatürden alınan birkaç standart örnek için deneysel sonuçlar verilecektir.

2. Karınca Algoritması

Karınca kolonileri iki nokta arasındaki en kısa yolu bulmada, yol üstüne yaydıkları fenomen sıvısıyla oluşturdukları basit haberleşme mekanizmasını kullanırlar.

Karıncaların yol seçimi, önceki karıncaların alternatif yollar üstüne bıraktıkları iz yoğunluğuna bağlıdır.



Şekil 1. Gerçek karıncaların davranışları

Şekil 1.a'da görülen A-E yolu üzerindeki karınca kolonisinin yoluna, Şekil 1b'de görüldüğü gibi bir engel konduğunda karıncalar H-B ve B-C yolları arasından seçim yaparken önceki karıncaların yola bıraktıkları fenomen miktarına bağlı karar vereceklerdir. Belirli bir zaman sonunda B-C yolunun H-B yolundan daha kısa olmasından dolayı B-C yolu üzerindeki fenomen miktarı daha yoğun olacaktır. Bu nedenle de karıncalar Şekil 1c'deki gibi daha kısa olan B-C yolunu tercih edecektir.

Karıncalar kolonilerinin yuvaları ile yemleri arasındaki bu davranışları kombinatoriyol optimizasyon problemlerinin çözümünde uygulanmıştır. Bunun için geliştirilen algoritmaların genel yapısı şu şekilde gösterilebilir (Talbi ve diğerleri, 2001);

ADIM-1 (Başlangıç)

- İlk Fenomen Miktarının Hesaplanması

ADIM-2 (İterasyonlar)

- Her bir karınca için tekrarla

- Fenomen izini kullanarak çözümün geliştirilmesi

- Fenomen miktarının güncellenmesi

- Durdurma kriteri sağlanana kadar

Bütün karıncalar başlangıç safhasında bir ilk çözüm oluştururlar. Oluşan bu çözümler fenomen izi miktarları göz önünde bulundurularak geliştirilirler. Her iterasyonun sonunda fenomen miktarı güncellenmelidir. Çünkü doğal ortamda karıncaların yolları üzerine bıraktıkları fenomen miktarı zamanla azalmaktadır. İterasyonlar önceden belirlenen durdurma şartı sağlanana kadar devam edecektir.

3. Kareli Atama Problemi için AntSimulated

KAP'nın çözüm bulmada kullanıldığı KAP önemli bir *NP-Hard kombinatoriyal optimizasyon problemler* sınıfıdır. İlk olarak Koopmans ve Beckman (1957) tarafından geliştirilen KAP'de $n \times n$ 'lik bir problem için $A = \{A_1, A_2, A_3, \dots, A_n\}$ yerlerin n setini, $B = \{B_1, B_2, B_3, \dots, B_n\}$ ise nesnelere n setini oluşturmaktadır. Bu durumda, m_{ij} A_i ile A_j arasındaki mesafeyi, a_{ij} ise B_i ile B_j arasındaki akışı ifade etmektedir.

$$\text{Problemin amaç fonksiyonu; } \min_{\pi \in \Pi} f(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \times m_{\pi_i \pi_j} \quad (1)$$

şeklinde ifade edilir. n unsurun permütasyonu $\Pi(n)$ ile ifade edilir ve amaç fonksiyonunun minimizasyonu amaçlanır.

3.1. Başlangıç

KAP'nin çözümü için geliştirdiğimiz AntSimulated algoritmasında öncelikle m adet karıncanın her birisi rastsal olarak bir ilk çözüm geliştirir. Geliştirilen m çözüme daha sonra bahsedilecek olan yerel arama optimizasyonu uygulanır. Elde edilen m adet yeni çözüm içerisinde en iyisi (π^*) alınarak Eşitlik-2'ye göre fenomen matrisinin başlangıç değerleri elde edilir. $i, j \in [1, 2, 3, \dots, n]$ için;

$$\tau_{ij}^0 = \frac{1}{100} \times f(\pi^*) \quad (2)$$

3.2. Çözümün Geliştirilmesi

Algoritmanın bu safhasında iki kademeli geliştirme yapılmıştır. Öncelikle fenomen matrisinde faydalanılacak daha sonra yerel arama algoritması kullanılacaktır.

Geliştirmenin ilk kademesinde her bir iterasyon için her bir karınca çözümüne T adet takas uygulanır. Takas işlemi ikili yer değiştirme şeklinde yapılmaktadır. Öncelikle rastsal olarak p elementi seçilir. Eşitlik-3'deki ifadeyi maksimum yapacak q değeri tespit edildikten sonra π çözüm permütasyonunda p ve q değerleri yer değiştirir.

$$\tau_{p\pi_q} + \tau_{q\pi_p} \quad (3)$$

Geliştirmenin ikinci kademesinde TB'e (Kirkpatrick ve diğerleri, 1983) dayalı bir yerel arama prosedürü kullanılmıştır. TB'nin arama prosesi içerisinde yerel bir minimuma ulaşmasına rağmen global minimum için arama yapması seçim sebebi olmuştur. Bu çalışmada, soğutma çizelgesi olarak Eşitlik-4 kullanılmıştır. Burada T_k mevcut iterasyonun sıcaklığı T_{k+1} ise bir sonraki iterasyonda kullanılacak sıcaklık değeridir.

$$T_{k+1} = T_k / 1 + (r \times T_k) \quad (4)$$

Eşitlik-4’de soğutma oranını ifade eden r değeri Eşitlik-5’e göre hesaplanmıştır.

$$r = [(T_0 - T_K) / (K - 1) T_0 T_K] \quad (5)$$

Burada ise T_0 ilk sıcaklığı, T_K son sıcaklığı ifade ederken K yerel arama prosesinin iterasyon sayısını verir.

Bu çalışmada yerel arama algoritması için başlangıç sıcaklığı 100, bitiş sıcaklığı 0,001 değerini almıştır. Durdurma kriteri olarak iterasyon sayısı (100) dikkate alınmıştır. Kabul olasılık fonksiyonu olarak Lee ve Wang’ın geliştirdiği kabul olasılık fonksiyonu kullanılmıştır (Lee ve diğerleri). Bu seçimlerin yapılmasında daha önceki başarılı uygulamalardan faydalanılmıştır.

3.3. Fenomen Matrisinin Güncellenmesi

Doğal yaşamda uçucu bir madde olan fenomen sıvısı zamanla azalma gösterir. Bu davranıştan esinlenerek algoritmanın her bir iterasyonunun sonunda fenomen miktarının güncellenmesi gerekir. İki farklı biçimde yapılan bu değişimin birincisinde, bütün iz değerleri belirli bir katsayıyla çarpılarak azaltılır (Eşitlik-6). İkinci güncelleme denkleminde ise mevcut iterasyon sonunda bulunan en iyi çözümde hesaba katılır (Eşitlik-7). Bu süreç önceden belirtilen durdurma kriteri sağlanana kadar devam edecektir.

4. Deneysel Sonuçları

Geliştirdiğimiz AntSimulated algoritmasını C++ dilinde kodlayarak QAP-Library’den (QAP-Lib) alınan 8 problem üzerinde deneysel çalışmasını yaptık. Parametre ayarlaması için gerekli olan deneysel tasarım aşamasını henüz yapmadığımız için algoritmada kullanılan parametreleri daha önceden iyi sonuçlar elde etmiş çalışmalardan aldık (Talbi ve diğerleri).

Algoritmanın parametrelerinden iterasyon sayısı, takas sayısı ve karınca sayısı deneysel çalışılan problemin n boyutuna göre belirlenmiştir. Buna göre algoritma, iterasyon sayısı 100, takas sayısı n/3, karınca sayısı ise n parametreleri ile çalıştırılmıştır.

Her bir problem için 10 defa çalıştırılan AntSimulated algoritması, ilgili problemler üzerinde çalışılmış diğer sezgisellerle karşılaştırılmıştır. Tablo-1’de gösterilen bu karşılaştırmalarda, bilinen en iyi değerlerle çözüm değerleri arasındaki farkın ortalama değerleri kullanılmıştır.

Tablo 1. AntSimulated ve Diğer Sezgisellerin Çözüm Kalitesi

Problem	Bilinen En İyi Değer	TTA	TB	GMM	MKA	AntSimulated
nug20	2570	0,911	0,070	0	0	0
nug30	6124	0,872	0,121	0,007	0,098	0,021
sco64	48498	0,861	0,095	0,092	0,129	0,042
tai30a	1818146	0,286	0,907	0,439	1,311	0,641
tai80a	13557864	0,467	0,995	0,796	2,689	0,665
bur26a	5426670	-	0,1411	0,0120	0	0
bur26h	7098658	-	0,1268	0,0003	0	0
kra30a	88900	2,0079	1,4657	0,1338	0,6299	0,3215

Karşılaştırmada kullanılan algoritmalar; Battiti ve Techiolli’nin Tepkisel Tabu Araması (TTA), Fleurent ve Ferland’ın Genetik Melez Metodu (GMM), Connolly’nin Tavlama Benzetimi (TB) ve Gambardella, Taillard ve Clorninin Melez Karınca Algoritması (MKA)’dır.

5. Sonuç ve Gelecek Çalışmalar

Bu çalışmada QAP problemleri için iyi sonuçlar veren bir algoritmanın geliştirilmesi amaçlanmıştır. AntSimulated olarak adlandırdığımız bu algoritma lokal arama prosesi Tavlama Benzetimi olan Karınca Algoritması tabanlıdır. Geliştirilen algoritma yapılan deneysel çalışmaların sonunda Tablo-1’de gösterildiği gibi bazı problemler için bilinen en iyi sonuçları elde etmiştir.

Gelecekte öncelikle parametre ayarlaması yapmak için deney tasarımı yapılacaktır. Bu şekilde şuan elde edilen sonuçlardan daha iyi sonuçların elde edilmesi amaçlanmaktadır. Daha ileride ise diğer optimizasyon problemleri için yeni uygulamaların yapılması düşünülmektedir.

Kaynaklar

- Bullnheimer, B., Hartl, R.F., Strauss, C.**, Applying the ant system to the vehicle routing problem, *Second Metaheuristics International Conference, MIC'97*, Sophia-Antipolis, France, 1997.
- Colorni A., Dorigo M., Maniezzo, V., Trubian, M.**, Ant system for job-shop scheduling, *JORBEL Belgian Journal of Operations Research Statistics and Computer Science* 34 (1) (1994) 39–53.
- Dorigo, M.**, Optimization, learning and natural algorithms, Politecnio di Milano, *Ph. D. Thesis*, Italy, 1992.
- Dorigo, M., Maniezzo, V., Colorni, A.**, The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Mans, and Cybernetics*, 1 (26) (1996) 29–41.
- Kirkpatrick, S., Gelatt, Jr., C. D., and Vecchi, M. P.**, Optimization by simulated annealing, *Science*, 220, 671-680, (1983).
- Lee, S. S., Wang, H-P.B.**, Modified simulated annealing for multiple-objective engineering design optimization, *Journal of Intelligent Manufacturing*, 3, 101-108, (1992).
- Talbi, E.G., Roux, O., Fonlupt, D., Robillard, D.**, Paralel Ant Colonies for the quadratic assignment problem, *Future Generation Computer Systems*, 17 (2002) 441-449.
- QAP-Lib:** www.opt.math.tu-graz.ac.at/qaplib/