

Karınca Kolonisi Algoritmasının Zaman Çizelgelemesi Üzerine: Bir Modellemesi ve Uygulaması

Hülya Özdağ¹, Nilgün Aygör¹, Aykut Parlak²

¹ Yıldız Teknik Üniversitesi Matematik Bölümü, İstanbul

² Yıldız Teknik Üniversitesi Matematik Mühendisliği Bölümü Öğrencisi, İstanbul

ozdag@yildiz.edu.tr, aygor@yildiz.edu.tr, aykut.parlak@gmail.com

Özet: Bu çalışmada NP-hard zaman çizelgeleme problemlerinden birine örnek olan ders dağılımı optimizasyon probleminin Karınca Kolonisi Algoritmasına (KKA) uygun ağaç graf modeli, öğretim üyelerinin gün ve saat talepleri doğrultusunda derecelendirilerek oluşturulmuş ve model üzerinde KKA ile, dersleri ve öğretim üyelerini kısıtlar doğrultusunda optimum atamaya yönelik bir uygulama geliştirilmiştir. Yapılan uygulamada Y.T.Ü. Matematik Mühendisliği bölümünün güz dönemi öğretim planı; öğretim üyelerinin saat ve gün talepleri baz alınarak, ağırlıklı graf modeli oluşturulmuş ve dağılım karınca kolonisi algoritması yardımı ile optimum şekilde gerçekleştirilmiştir. Bu uygulama C programlama dili ile geliştirilmiş daha sonra veri girişi için ara yüz gereksiniminden dolayı Python programa dili ile yeniden yazılmıştır. Ayrıca verilerin kayıt yapılması ve sorgulanması için MySQL veritabanı kullanılmıştır.

Anahtar Sözcükler: Zaman çizelgesi, Ağırlıklı Graf, Karınca Kolonisi Algoritması, Yapay Zeka

Modeling and an Application on Ant Colony Algorithm Based on Time Schedule

Abstract: In this study, graph tree model of curriculum which is a NP-hard time scheduling optimization problem is designed base on Ant Colony Algorithm with weighted according to the demands of lectures' days and hours. Also an application made on this model based on Ant Colony Algorithm to build curriculum of Y.T.U. Mathematical Engineering department with lecturers' restrictions. This study is especially on algorithm and modeling. The application was written in C language but after it was wrote in Python language because of interface needed for data enter. What is more, MySQL database used to insert and investigate the data.

Keywords: Timetabling, Weighted Graf, Ant Colony Algorithm, Artificial Intelligence

1.Giriş

NP-hard zaman çizelgeleme problemlerinden bir tanesi de üniversite öğretim planının oluşturulmasıdır[1]. Öğretim planı; derslikler, sınıf programı, öğretim üyeleri gibi parametreler baz alınarak hazırlanmaktadır. Ancak öğrenciye ve öğretim üyelerine uygunluğu, pedagojik özelliği gibi göreceli parametrelerin dahil edilmediği bir plan eksik bir plandır. Bu çalışmada amaçlanan, göreceli değerlerin öğretim planı oluşturulurken, parametre olarak

dahil edilmesidir. Ancak bu parametrelerin de dahil edilmesiyle çözüm uzayı daha da büyümektedir. Çözüm uzayının büyük olduğu optimizasyon problemlerinde, çözüm uzayının bir kısmını tarayıp, optimal veya optimala yakın bir çözüm bulan algoritmalara *sezgisel algoritmalar* denir[4]. Bu algoritmalar genel olarak geri beslemeli çalışmaktadırlar.

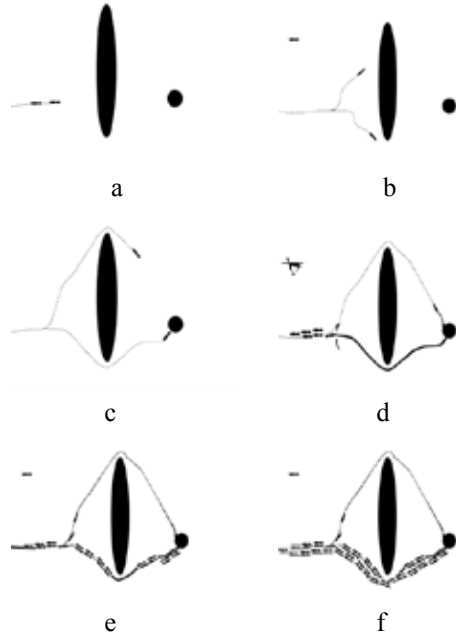
Çözülecek problemin bir amaç fonksiyonu vardır ve bir sonraki iterasyon ile elde edilecek nokta bir önceki noktanın amaç fonksiyono-

nunda aldığı değere göre belirlenmektedir. Bu çalışmada öğretim planı, öğretim üyelerinin zaman talepleri derecelendirilerek, sezgisel algoritmalarla sürü zekasına dayanan “Karınca Kolonisi Algoritması” üzerinde modellenmiş ve Python diliyle yazılan bir uygulama ile göreceli parametre dahil edilerek öğretim planı oluşturulmuştur.

1.1 Biyolojik Altyapı

Karıncalar tek başlarına basit birer canlı olmalarına karşın koloni şeklinde yaşamaları onlara yetenekleri üstünde bir güç katmaktadır. Görme duyularının kötü olmasına karşın kimyasal algılayıcıları gelişmiştir. Karıncalar gündelik hayatta yiyecek ararken yollarını ve rotalarını kaybetmemek, ayrıca diğer karıncalar ile iletişim sağlamak için geçtikleri yollara “feromon” adında bir tür hormon salgısı bırakırlar. Bir karınca yiyecek bulduğunda diğer karıncalarında yiyeceğe ulaşabilmesi için dönüş yoluna da feromon salgılar. Böylece diğer karıncalar da izlerdeki feromon yoğunluğuna göre izleri takip ederek yiyeceğe ulaşabilirler. Bu sistem ile karıncalar diğer karıncaların deneyimlerini kullanarak optimal yolu ve yiyeceği bulurlar. [2]

Örneğin yiyecek arayan(Şekil 1a) karıncalar, üzerinde iz olmayan bir yolda engelle karşılaştıklarında, rassal bir şekilde bazılarının uzun yolu, bazılarının ise kısa yolu seçtiğini düşünelim(Şekil 1 b). Kısa yolu seçen karıncalar yiyeceğe daha çabuk ulaşacak ve geldiği yoldan feromon salgılayarak geri döneceklerdir. Böylece kısa yolda daha fazla feromon birikecek(Şekil 1 d) ve arkadan gelen karıncalar feromon oranı fazla olan yolu daha fazla tercih edeceklerdir. (Şekil 1 e). Dolayısıyla feromon oranı arttıkça seçilme şansı artan, seçildikçe de feromon oranı artan kısa yol bulunmuş olacaktır(Şekil 1 f). Feromon uçucu bir madde olduğu için zamanla izler kaybolur. Bundan dolayı besine ulaşmayan yollardan da izler silinir ve böylece karınca yanılmaktan kurtulur. [2]



Şekil 1.

2. Karınca Kolonisi Algoritması

Karınca kolonisi algoritması (KKA) karıncaların sürü zekâsını temel prensip almaktadır. Bir popülasyon algoritması olan KKA, problemin muhtemel çözüm noktalarından ve ağırlıklı ayrıtlardan oluşan graf modeli üzerinde, sonlu sayıda karıncayı, amaç fonksiyonunu baz alan feromon salgısı ile dolaştırıp, olayı simüle etmektedir. Böylece sürü zekası ile optimum çözüme yakın çözüm kümesi elde edilmektedir. KKA geri beslemeli olarak çalışır. Oluşturulan modelde, karıncaların aramaları problemin çözüm kümesini, yiyeceğe ulaşan yol amaç fonksiyonunu, feromon madde ise hafıza kavramını temsil eder.

Karınca kolonilerinin davranışlarını esas alan ilk algoritma Marco Dorigo tarafından 1996 yılında ortaya atılmıştır.[3]

Bu sistemde gerçek karıncaların birtakım özellikleri değiştirilerek yapay karıncalar tasarlanmıştır. Ancak problemin KKA ‘ya uygulanmasının bazı özellikleri vardır. Problem, tepe ve

ayrıtların olduğu bir yol şeklinde modellenir. Bu yollar arasında karınca, aşağıda tanımı verilen olasılık fonksiyonunun değerlerine göre karar vermektedir.

i noktasında bulunan k karıncasının olası gidebileceği yollar kümesi nin içinden, j yolunun seçilme olasılığı fonksiyonu:

i noktasında bulunan k karıncasının olası gidebileceği yollar kümesi S^p nin içinden, j yolunun seçilme olasılığı fonksiyonu:

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{c_{ij} \in N(s^p)} \tau_{ij}^\alpha \cdot \eta_{ij}^\beta} & c_{ij} \in N(s^p) \text{ ise,} \\ 0 & \text{aksi halde,} \end{cases} \quad (1)$$

şeklinde dir.

$N(s^p) : l, k$ karıncası tarafından uğranmamış bir nokta olarak, (i, l) ayrıtlarından uygun olanlarının kümesidir.

$\tau_{ij} : (i, j)$ köşelerindeki feromon iz miktarıdır.

Feromon iz miktarı (2) numaralı bağıntıya göre güncellenir.

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

ρ : Feromon izinin buharlaşma oranı. $(0 < \rho < 1)$

m : Toplam karınca sayısı.

$\Delta \tau_{ij}^k$: k karıncasının (i, j) ayrıtlına bıraktığı feromon miktarını göstermektedir. Bu miktar (3) numaralı bağıntıya göre hesaplanır.

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{eğer } (i, j) \text{ ayrıtlı } k \text{ karıncası turunda ise,} \\ 0 & \text{aksi taktirde,} \end{cases} \quad (3)$$

Q : Sabit bir değer.

L_k : k ıncı karınca tarafından taranan tur uzunluğudur.

$\eta_{ij} : (i, j)$ köşeleri arasındaki görünürlük (visibility) veya bağlantının amaç fonksiyonu özelliğidir.

$$\eta_{ij} = 1 / d_{ij}$$

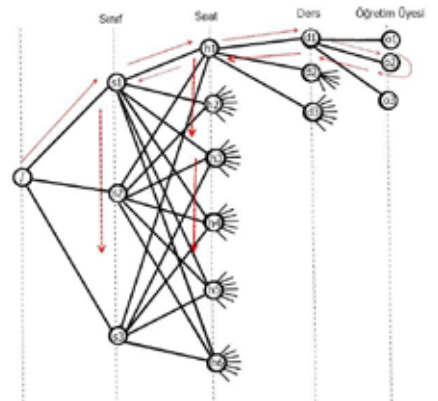
d_{ij} : i ve j noktaları arasındaki uzaklıktır.

α : Feromon izine verilen önemi gösteren parametredir.

β : Problemd e görünürlük(visibility) değerine yani amaç fonksiyonunun uygunluğuna verilen önemi gösteren bir parametredir. [3]

3. Ders Çizelgeleme Probleminin Karınca Kolonisi Algoritması ile Modellenmesi

Ders çizelgeleme probleminin Karınca Kolonisi Algoritması ile çözüleb ilmesi için tepeler ve ayrıtlardan oluşan algoritmaya uygun graf modeli oluşturulur. Bu çalışmada tepeler; sınıf, saat, ders ve öğretim üyelerinden oluşmaktadır. Ayrıtlar ise sınıf-saat, saat-ders, ders-öğretim üyesi arasındadır. Böylece her bir sınıf programı için ağaç graf modeli elde edilmiş olur. Ders atama problemine öncelikle sınıf seçilerek başlanır. Daha sonra bu sınıf için atama yapılacak bir saat seçilir, bu saat için bir ders ve en sonunda da bu derse bir öğretim üyesi atanır. Bu döngü, ders saatlerine atama işlemi bitinceye kadar devam eder. Daha sonra sınıflara geçilerek aynı işlemler uygulanır. Bu yaklaşım ile (Şekil 2)'de gösterildiği gibi ağaç grafi elde edilmiştir. Model oluşturulduktan sonra, amaç fonksiyonu doğrultusunda karınca kolonisi algoritması ile optimizasyon yapılabilir.



Şekil 2.

3.1. Karınca Kolonisi Algoritmasının Uygulaması

Program iki ana bölümden oluşur. Birinci kısım, verilerin programda kullanılmak üzere, veri tabanına girilmesi için ara yüzlerden ve

veritabanından, ikinci kısım ise girilen bilgiler doğrultusunda ders dağılımını yapan ve çıktı olarak bir text dosyası oluşturan ana programdan oluşmaktadır.

3.1.1. Veri Tabanı Oluşturmak İçin Ara Yüzler

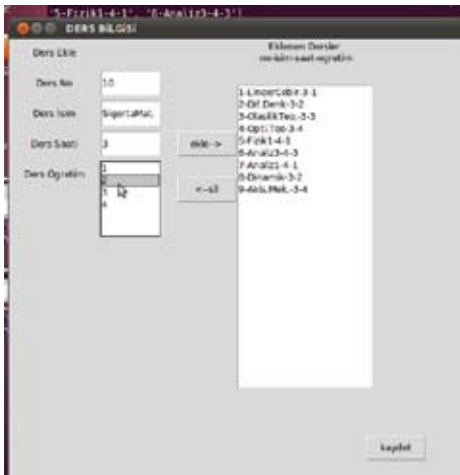
Programda üç ana tabloya ihtiyaç duyulmaktadır. Bunlar sırası ile; sınıfların ders programlarını içeren program tablosu, ders bilgilerinin olduğu ders tablosu ve öğretim üyelerinin verdiği derslerin ağırlıklı zaman bilgisini içeren ders tablosudur. Her biri için ayrı ara yüzler oluşturulmuştur.

3.1.1.1. Sınıf programı tablosu

Bu tabloda sınıfların öğretime açık olduğu gün ve saatler tanımlanarak kaydedilir. Aşağıdaki ara yüzde açık saatler için 1, kapalı saatler için 0 girilerek, oluşturulan sınıf programı veritabanına kaydedilir.



Şekil 3. Sınıf Programı Tablosu



Şekil 4. Ders Tablosu

3.1.1.2. Ders tablosu

Bu tabloda derslerin özellikleri tanımlanmaktadır. Aşağıdaki ara yüz kullanılarak dersin kodu, adı, saati ve sınıf bilgileri girilerek veritabanına kaydedilir.

3.1.1.3. Öğretim Üyesi Tablosu

Bu tabloyla öğretim üyelerinin vermek istedikleri dersler, gün ve saat talepleri doğrultusunda istek derecelendirmesi yapılarak veri tabanına kaydedilir.



Şekil 5.

3.1.2. Ana Programın Çalışması

Bütün saatler ve sınıflar taranarak oluşan tam bir atama işlemi, programın “karınca” diye adlandırılan temel elemanı tarafından yapılmaktadır. Her karınca, model ağaç üzerinde bir ders atama işlemi gerçekleştirir. Karıncalar, atama işlemini ilgili ayrıtlarda bulunan feromon miktarları ve ağırlıkları temel alarak yaparlar. Karınca, atama işlemini tamamladığında, modelin kısıtları doğrultusunda geçtiği ayrıtların ilgili bölümüne, amaç fonksiyon değerini temel alan feromon değerini bırakır. Bir karıncanın izlediği basamaklar aşağıdaki gibidir:

- Bir sınıf seçilir, bu sınıfın boş olan bir saati seçilir.
- Bu saate, KKA prensibine göre bir ders atanır.
- Atanmaz ise boş saat olarak kaydedildikten sonra başa dönerek boş olan başka bir saat seçilip işleme devam edilir.
- Ders belirlendikten sonra KKA prensibine göre öğretim üyesi atanır.

- Atanmaz ise boş saat olarak kaydedilip tekrar başa dönerek boş olan başka bir saat seçilir ve işleme bu şekilde devam edilir.
- Sınıfın tüm saatleri işlem gördükten sonra diğer sınıflara geçilir.
- Tüm sınıflar tamamlandıktan sonra amaç fonksiyon değeri(KKA içerisinde ki L_k değeri) hesaplanır.
- Kayıtlı optimum değere göre daha iyi bir değere sahip ise önceki kayıt silinerek yeni değer ve ders programı kaydedilir.
- Geçici feromon listesi, program listesi, öğretim üyesi programı güncellenir. Dönü diğer bir karıncaya geçer.

3.2. Programın çalışması

Sırası ile program bilgisi, ders bilgisi, öğretim üyeleri bilgisi ara yüzler ile veritabanına alınır. Daha sonra ana program çalıştırılır. Programda beş parametre kullanılmaktadır. Bunlar: KKA'nın bir bileşeni olan a , $1/2$, p , iterasyon ve karınca sayılandır. Program veri tabanına bağlanıp veri tablolarını kendi içine kopyalar. KKA'nın çalışması için eşit başlangıç değerleri ile feromon listesi oluşturulur. Optimal noktaya çok uzak bir amaç fonksiyon değeri belirlenir. İterasyon döngüsü başlar. Her iterasyon sonunda genel feromon listesine eklenmek üzere geçici bir feromon listesi oluşturulur. Karınca döngüsü başlar. Tüm karıncalar dolaşımını tamamladıktan sonra genel liste, geçici feromon listesi ile güncellenir. İterasyon tamamlanana kadar program devam eder. En sonunda geçerli program ve değer bir text dosyası olarak sunulur.

Y.T.Ü Matematik Mühendisliği bölümü baz alınarak yapılan atamalarda amaç fonksiyonunun değeri, iterasyon ve karınca turlarına göre doğrusal olmasa da sürekli daha iyi bir değer elde ederek devam etmektedir.

4. Sonuç

Bu çalışmada öğretim planının, bir parametreye bağlı derecelendirme ile Karınca Kolonisi Algoritması'na uygun ağırlıklı graf modeli geliştirilmiş ve bu modelin uygulanabilirliğini gösteren bilgisayar programı ile de desteklenmiştir. Ayrıca model, bir çok ağırlıklandırma parametresi ve amaç fonksiyonu değişkeni eklenerek geliştirilmeye de açıktır. Bu doğrultuda bilgisayar programına bir çok modül eklenerek karmaşık atamalar yapılabilir. Diğer taraftan bu model ve uygulaması ders çizelgelemenin dışında bir çok probleme de uyarlanmaya açıktır.

Kaynaklar

- [1] Çoruhlu Alptekin, Sınav Personel Çizelgeleme Mode- li, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, ANKARA (2007)
- [2] Goss. S., Aron. S., Deneubourg J.L. and J.M. Pasteels, Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* 76, 579-581 (1989)
- [3] Dorigo Marco, Mauro Birattari, and Thomas Stützle Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique, IRIDIA-Technical Report Series, Technical Report No.TR/IRIDIA/2006-023 (2006)
- [4] Natallia Kokash. "An introduction to heuristic algorithms," URL: <http://dit.unitn.it/~kokash/documents/Heuristic-al> <http://dit.unitn.it/~kokash/documents/Heuristical>. (2005)