

ARI KOLONİSİ OPTİMİZASYON ALGORİTMASI KULLANARAK ŞOFÖR- HAT-ZAMAN ÇİZELGELEME

Mustafa Servet KIRAN¹, Mesut GÜNDÜZ¹

¹Bilgisayar Mühendisliği Bölümü, Mühendislik-Mimarlık Fakültesi, Selçuk
Üniversitesi, 42075 Selçuklu, KONYA
mskiran@selcuk.edu.tr, mgunduz@selcuk.edu.tr

Özet

Bu çalışmadaki şoför-hat-zaman çizelgeleme problemi şehir içi hatlarda çalışan toplu taşıma araçlarının seferlerinin şoförler için çizelgenmesini kapsar. Genellikle birçok hatta ait birçok sefer aynı istasyondan başlamak suretiyle hattın güzergâhında halka şeklinde icra edilmektedir. Yolcular duraklarda bulunan listeler yardımıyla bekledikleri otobüslerin ne zaman geleceğini tespit ederler. Bu listeler yolcuların gün içindeki yoğunluk durumu düşünülerek hazırlanır ve ayrıca listelerdeki seferleri icra edecek şoförler içinde çizelgenmesi gerekir. Seferlerin şoförler için çizelgenmesi hali hâzırda istasyon sorumlusu tarafından yürütülmektedir. Sefer sayısının fazla olması durumunda çözüm uzayı genişlemesi istasyon sorumlusunun olası en iyi çözümleri gözden kaçırmamasına neden olur. Bu çalışmada Arı Kolonisi Optimizasyon algoritması hatta ait seferlerin şoförler için çizelgenmesi amacıyla kullanılmıştır. Sonuçlar hâli hazırda kullanılmakta olan çözüm ile kıyaslanmıştır. Önerilen yöntem ile elde edilen sonuçlar şoför sayısı, dinlenme süresi, yemek molası kriterlerine göre hâli hazırda kullanılan çözümden daha iyidir

Anahtar Kelimeler: Şoför-hat-zaman çizelgeleme, arı kolonisi optimizasyonu, şehir içi yolcu taşımacılığı.

DRIVER-LINE-TIME SCHEDULING WITH BEE COLONY OPTIMIZATION ALGORITHM

Abstract

Driver-line-time scheduling problem contains scheduling of expeditions of urban public transportation for drivers. A lot of expeditions are generally performed as ring on route of line by starting same station. The expeditions, which are in a list, are prepared by considering density of travelers during the day. Travelers learn when the bus will come by looking to the list of expeditions. Moreover, these expeditions must be scheduled to the drivers. Scheduling process is made by chief of station in current situation. If expeditions in the lists are over numbered, the process of scheduling which is made by the station chief gets hard. In this work, Bee Colony Optimization Algorithm is used for being scheduled of expedition times. Results obtained by the proposed method are compared with the solution prepared by the station chief. The proposed method is better than the currently when the rest-period, lunch break for the drivers and number of drivers who they perform the expeditions are considered.

Keywords: Driver-line-time scheduling, bee colony optimization, urban public transportation.

1. Giriş

Gerçek dünyada şehir içi yolcu taşımacılığı için kullanılan otobüs sefer sayısı, genellikle yolcuların gün içindeki yoğunluk durumu düşünülerek belirlenmektedir. Seferler genelde halka şeklinde icra edilmektedir. Herhangi bir hatta bağlı olan seferlerin kendilerine ait istasyondan ve varış noktasından kalkış saatleri vardır. Yolcuların duraklarda fazla beklememesi için bu kalkış saatlerinde mutlaka birer araç istasyondan hareket etmelidir. Yani yolcular için hazırlanan seferlerin ayrıca şoförler için de çizelgelenmesi gereklidir.

Seferler çizelgelenirken şoförlerin çalışma süresi, dinlenme süresi, yemek molası gibi kısıtlara uyulması zorunludur.

Çalışma Süresi: Şoförün gün içindeki çalışma süresi kanunlarla belirlenmiş olan mesai saatini aşmamalıdır. Aksi takdirde şoföre fazla mesai ücreti ödenmektedir. Fazla mesai ücreti de belirli bir saati geçmemelidir.

Dinlenme Süresi: Her şoför bir seferi icra ettikten sonra diğer sefere geçmeden önce bir müddet dinlenmelidir. Genellikle bu süre 10 dakika olarak verilmektedir.

Yemek Molası: Mesai saatleri arasında şoföre yemek yemesi için verilmesi gereken süredir. Genellikle bu süre 60 dakikadır.

Yukarıdaki kısıtlara uyarak tüm seferlerin minimum sayıda şoför ile icra edilmesi gerekir. Görüldüğü gibi problem, çizelgelemeyi ve şoför sayısının minimizasyonunu birlikte barındırır. Çizelgelenecek sefer sayısı arttıkça çizelgeleme ve optimizasyon süreçleri de karmaşıklaşmakta ve zorlaşmaktadır.

Yolcular için hazırlanmış olan sefer listesinde birden fazla sefer birden fazla hatta bağlıdır ve bir listede tutulmaktadır. Halen bu listedeki seferlerin şoförler için çizelgenmesi manuel olarak istasyon sorumlusu tarafından yapılmaktadır. Hazırlanan çizelgenin optimizasyonu ise problemin zorluğundan ve karmaşıklığından dolayı istasyon sorumlusu tarafından yapılamamaktadır. Genellikle çizelgelemenin bir insan tarafından yapılması olası en iyi sonuçların kaçırılmasına yol açabilir. Ayrıca listedeki sefer sayısının artırılması veya azaltılması tekrar çizelgelemeyi beraberinde getirir. Bu da zaman ve verim kaybına yol açar.

Yukarıda bahsedildiği gibi bu problemin herhangi bir bilgisayar destekli çizelgeleme veya optimizasyon yöntemiyle çözülmemiş olması elde edilen sonuçların optimuma ne kadar yakın olduğunu değerlendirmeye imkan vermemektedir.

Bu çalışmada seferlerin çizelgenmesi ve optimizasyon süreçleri için Arı Kolonisi Optimizasyon (Bee Colony Optimization- BCO) algoritması kullanılmıştır. Bu algoritma probleme uyarlama kolaylığı, çalışma sürecinde çevresel değişikliklere hızlı tepkiler verebilme yeteneği, kolay anlaşılabilirliği ve basit mantığı nedeniyle oldukça popüler bir hale gelmiştir. Örneğin gezgin satıcı problemi [1][2][3], internet sunucularının yerleşimi [4], ride-matching probleminin çözümü [5], makine-görev (job shop) çizelgeleme [6] ve birçok benzer problemler arıların davranışlarından esinlenilerek ortaya konulmuş yöntemler ile çözülmüştür.

Bu çalışmanın 1. bölümünde giriş ve problemin tanıtımı yapılmıştır. 2. bölümde problem detaylandırılmıştır. 3. bölümde çözüm için uygulanan BCO algoritması

sunulmuştur. Deneysel sonuçlar 4. bölümde, bulguların tartışılması ise 5. bölümde verilmiştir.

2. Problem Arka Planı

Bu problem şehir içi yolcu taşımacılığındaki otobüs seferlerinin şoförlere çizelgelenmesini kapsar. Şehir içi yolcu taşımacılığında otobüslerin çeşitli güzergahları vardır ve bu güzergahlarında seferleri bulunmaktadır. Her seferin istasyondan ve varış noktasından kalkış saati vardır. Seferler halka şeklindedir ve seferin zamansal uzunluğu bir seferin istasyondan ve varış noktasından kalkış saatinin farkının iki katıdır. Bir şoför için çizelgelenen seferlerin süresi mesai saatini geçmemelidir. Fazla çalışmalar için ek mesai ücreti ödenmektedir[7].

Sefer saati, güzergah numarası gibi bilgiler bir listede tutulmaktadır. Duraklardaki yolcular bu liste yardımıyla bekledikleri aracın ne zaman geleceğini bilirler. Listede bulunan bilgiler Şekil 1’de gösterilmektedir.

Güzergahın adı	GAZİ-D.PINAR ALAADDİN		
	GAZİ CAD.-DUMLUPINAR- AYDINLIK-SANAYİ-ADESE- BELEDİYE-ALAADDİN		Güzergah bilgisi
Güzergah no	53		
Durak	CUM'TEN	ALAADDİN	Varış noktası
	06.00	06.40	
	06.20	07.05	
	06.35	07.20	
	06.50	07.35	
	07.00	07.45	
	07.10	07.55	
	07.20	08.05	
	07.30	08.15	
Duraktan kalkış saati			Varış noktasından kalkış saati

Şekil 1. Örnek bir hatta ait bilgiler

Şekil 1’e göre Cumhuriyet (Cum) istasyonundan saat 06:00’da kalkan bir araç varış noktası olan Alaaddin’e 40 dakikada varmalıdır. Aynı araç varış noktasından 06:40’da hareket ederek istasyona 40 dakikada dönmelidir. Bu seferin süresi kalkış

saatlerinin farkının iki katı olan 80 dakika olarak bulunur. Bulunan süreye 10 dakikalık dinlenme süresi eklendiğinde şoförün seferi tamamladıktan sonra yeni bir sefere hazır olması 90 dakika zaman alır. Kısacası 06:00'daki seferi icra eden şoför ikinci olarak 07:30'daki seferi icra edebilir. 07:30'da Cumhuriyet'te birden fazla sefer olduğunda hangisinin ikinci sefer olarak tayin edilmesinin bulunması problemin kombinasyonel bir hal almasına sebep olur. Ayrıca çözümü kısıtlayan bir diğer etmen de yemek süresidir. Herhangi bir şoför 3 veya 4 saatlik bir çalışmadan sonra yemek molası vermelidir. Moladan önceki son seferden döndükten sonra 60 dakikalık bir yemek molası verilir. Yemek molası bitiminde şoföre tayin edilecek sefer yemek molasının bitimine en yakın olmalıdır.

Yukarıda gösterilen şekilde devam eden bir günlük çalışma periyodu yemek molası dahil 8,5 saati geçmemelidir. Çizelgeleme sürecinin bitmesi için çizelgelenmesi gereken hiçbir sefer kalmamalıdır ve çizelgelenmesine başlanan şoförün mesaisi dolmalıdır. Çizelgelemesinin sona kaldığı seferlerde normal olarak şoförün mesaisi dolmamakta ve çözüm tamamlanamamaktadır. Çözümün sonlandırılması için bu duruma düşen şoförlere joker sefer ataması vardır. Joker sefere denk gelen her şoförün mesaisi dolduğundan çözüm sonlanabilmektedir. Ayrıca aynı istasyona bağlı hatların bazılarının kalkış yerleri farklıdır. Kalkış yerleri arasındaki mesafenin fazladan zaman ve yakıt sarfiyatına neden olmasından dolayı farklı kalkış yerlerine sahip hatlar gruplanarak çizelgeleme yapılmalıdır.

3. Problem için Uyarlanmış Arı Kolonisi Optimizasyon Algoritması

BCO algoritması gerçek arıların yem arama davranışını ve bilgi paylaşımı için yapılan sallanım dansını problemin çözümü için simüle eder [8]. BCO algoritması yalnız bir algoritmadır ve temelde iki aşamadan oluşur. İleri geçişlerde tüm arılar işçi arıyken geri geçiş aşamasında bazı arılar gözcü bazı arılar da işveren (recruiter) arı olur. Bir arının işçi veya gözcü arı olması arının çözümüne olan sadakatine bağlıdır. Eğer arı bulunduğu kaynağa sadık ise bu arı işçi arı, değilse gözcü arıdır. İleri geçiş yem toplanmasının, geri geçiş sallanım dansının yani bilgi paylaşımının modellenmesidir. BCO'da ileri geçiş aşamasında sadece işçi arılar vardır ve bu arılar geri geçiş aşamasında ya gözcü arı olur ve kendisine yeni bir yiyecek kaynağı atanmasını bekler ya da bulunduğu kaynağı iyileştirmeye çalışan işçi arı olarak yem toplamaya devam eder

[9]. Ayrıca gerçek arılar da olduğu gibi yapay arıların da belli bir seviyede hafızası vardır.

3.1. Nektar toplama

BCO algoritmasında işçi arı bulduğu nektar kaynağından faydalanmaya hemen başlar. Keşif ve faydalanma algoritmanın çözüm uzayını araştırmasıdır ve ileri geçiş esnasında yapılır. Her bir ileri geçişte arıların seçeceği çözüm, seçilebilecek çözümlerin değerlerine bağlı olarak rulet tekerleği prensibine göre yapılır.

Arının seçeceği yola ait ihtimal değeri (1) formülü ile hesaplanır.

$$p_k(t) = 1 - \frac{d_k(t)}{\sum_{k=1}^n d_k(t)} \quad (k=1 \dots n, \text{ seçilebilecek seferler}) \quad (1)$$

Formülde t çevriminde seçilebilecek mümkün olan seferlerden birinin ihtimali p_k , arının mevcut sefere olan zamansal uzaklığı d_k 'dır.

3.2. Yapay arılar arasındaki bilgi paylaşımı

Haberleşme neticesinde koloni bireyleri birbirlerinin kaynakları hakkında bilgi sahibi olurlar. Geri geçiş esnasında yapılan haberleşme sonucu bazı arılar kendi kaynaklarını terk ederek diğer işçi arıların çözümlerini iyileştirmeye çalışırlar. Arıların kendi kaynaklarını terk edip etmemesi kendi kaynaklarına olan sadakatlerine bağlıdır. Her arının sadakati ürettiği kısmî çözümün değerine göre hesaplanır.

Amaç fonksiyon (2) formülü ile ve her arının kendi çözümüne olan sadakati (3) formülü ile hesaplanmaktadır.

$$OFV_j = \sum_{i=1}^n B_{j,i} \quad (j=1 \dots n, \text{ ileri geçişteki arı sayısı}) \quad (2)$$

(i=1...m, her arının aldığı sefer sayısı)

Bu formülde OFV_j , j. arının amaç fonksiyon değeridir. $B_{j,i}$ ise j. arının bulduğu i. çözümün değerini ifade eder.

$$L_i(t) = 1 - \frac{B_{j,OFV}(t) - OFV_{MIN}(t)}{OFV_{MAX}(t) - OFV_{MIN}(t)} \quad (3)$$

Formüldeki t herhangi bir çevrim olmak üzere;

$L_i(t)$: t anında i. arının kendi çözümüne olan sadakatini;

$B_{j,OFV}(t)$: t anında j. arının amaç fonksiyon değerini;

$OFV_{MIN}(t)$: t anında koloninin en küçük amaç fonksiyon değerini (En kötü kısmî çözümü);

$OFV_{MAX}(t)$: t anında koloninin en büyük amaç fonksiyon değerini (En iyi kısmî çözümü) ifade eder.

Geri geçiş sürecinde L_i değerleri 0 ile 1 arasında üretilen rastgele bir sayı ile test edilir. Üretilen rastgele sayıdan büyük olan sadakat değerleri işçi arıların kendi çözümlerini geliştirmeye devam edeceğini, küçük sadakat değerleri ise bu değere sahip işçi arıların gözcü arı olacağını gösterir. Bu formül, herhangi bir haberleşme süreci sonunda en az bir tane işçi arının kendi çözümünü terk edeceği, en az bir tane işçi arının kendi çözümüne sadık kalacağı ve koloninin diğer bireylerinin de bu aralıkta olacağını ifade eder.

Haberleşme sürecinde oluşan gözcü arıların hangi işçi arıların kısmî çözümüne atanacağı genellikle amaç fonksiyon değerlerinin yerleştirildiği rulet tekerleği prensibine göre belirlenir.

İşçi arıların daha iyi amaç fonksiyon değeri iki faydayı beraberinde getirir; bunlardan birincisi daha yüksek sadakat değeri ve kendi çözümüne olan bağlılığı, ikincisi ise daha fazla gözcü arı tarafından seçilme ihtimalidir.

3.3. BCO algoritmasının şoför-hat-zaman problemine uygulanması

BCO algoritmasının şoför-hat-zaman problemine uygulanabilmesi için şoförleri yapay arılar, seferleri de yapay arıların uğrayabileceği düğüm olarak ele almak yeterlidir. Ek olarak çevrim sayısına (N) tasarımcı tarafından karar verilir. Bunlara dayanarak BCO algoritmasının şoför-hat-zaman problemine uyarlanmış hali aşağıda verilmiştir:

1. Başlatma (tanımlar vb.)
2. $I=1$.
3. Yapay arıların sefer bilgilerini temizle.
4. Sefer listesinin ilk elemanını hafızaya al.
5. İleri Geçiş:
 - Mümkün olan seferleri değerlendir.
 - Rulet tekerleği prensibine göre bir tane sefer seç.

6. Geri Geçiş:

- Sallanım dansı ile bilgileri paylaş
- Sadakat testi ile gözcü ve işçi arıları belirle.
- Gözcü arılara işçi arılardan bir tanesinin kısmî çözümünü ata.

7. Çözüm tamamlanmadıysa İleri Geçiş'e (5. adıma git) devam et.

8. Koloni'deki en iyi çözüm global en iyiden daha iyiye global en iyi olarak kaydet.

9. $I=I+1$, $I<N$ ise 3. adıma git.

10. Global en iyiyi kaydet ve sefer listesinden çıkar.

11. Durma kriteri sağlanmadıysa 2. adıma git.

12. Bulunan en iyi çözümleri yaz.

Öncelikle kovanın bulunduğu nokta sefer listesinin ilk elemanına konumlandırılır. Daha sonra yapay arılar kovanın bulunduğu noktadan seçilebilecek tüm seferleri değerlendirerek kendilerine bir sefer tayin ederler. Tayin edilen seferle birlikte kısmî çözümün amaç fonksiyon değeri bulunur. Arıların amaç fonksiyon değerlerine göre kısmî çözümlerine olan sadakat değerleri hesaplanır. Hesaplanan bu değer 0 ile 1 arasında rastgele bir sayı ile test edilir (sadakat testi) ve arının işçi veya gözcü arı olacağına karar verilir. Test sonucunda oluşan gözcü arılara için işçi arıların kısmî çözümlerinden bir tanesi atanır. Çözüm tamamlanmadıysa yani çalışma süresi dolmadıysa tekrar ileri geçişe devam edilir. İleri geçiş bittikten sonra kovadaki en iyi çözüm global en iyiden daha iyiye saklanır. Bu işlemler N defa sürdürülür. Bu algoritma için durma kriteri ise çizelgelenmesi gereken sefer kalmamasıdır.

4. Deneysel Sonuçlar

Önerin yöntem, Konya Büyükşehir Belediyesi Cumhuriyet Otobüs İstasyonu'na bağlı seferlerin çizelgelenmesi için çalıştırılmıştır. Optimum çizelgenin olmaması nedeniyle elde edilen sonuçlar, istasyonda hâli hazırda kullanılan çizelgeye göre değerlendirilmiştir. Hâli hazırda kullanılan çizelgede şoför sayısını azaltmak amacıyla bazı durumlarda yemek molası ve dinlenme süresi şoförlere verilememektedir. BCO algoritması ile üretilen çizelgede seferler için gereken şoför sayısı azaltılmış, seferler arasında 10 dakikalık dinlenme süresi ve mesai saati içinde 60 dakikalık yemek molası kesinlikle her şoföre verilmiştir.

Simulasyon sürecinde 50 arı bir şoför için gereken çizelgeyi oluşturmak için çalıştırılmıştır. Seferlerin tümü çizelgeleninceye kadar BCO algoritması koşturulmuştur. BCO algoritması şoför-hat-zaman çizelgeleme işlemi için 50 defa koşturulmuş ve elde edilen sonuçların ortalaması Tablo1’de verilmiştir.

Tablo 1. Uygulama sonucunda elde edilen ortalama değerler

Kalkış Yeri	Sefer Sayısı	Hatalı Çizelgelenen Sefer Sayısı (%)	Ortalama Gereken Şoför Sayısı	Başarı (%)
1	77	5.45	7.50	94.55
2	85	6.59	9.10	93.41
3	269	9.28	55.96	90.72

Tablo 1’e göre bir istasyona bağlı 3 kalkış yerinden hareket eden 431 seferin %92.89’u yaklaşık 73 şoför ile çizelgelenmiştir. Çizelgelenemeyen seferlerin şoförlere ek mesai ücreti ödenerek çizelgelenmesi mümkündür.

Elde edilen sonuçlar ile hâli hazırda şoförler tarafından kullanılmakta olan çizelge karşılaştırıldığında BCO ile üretilen çizelgede şoför sayısı 6 azaltılmıştır ve tüm şoförler seferler arasında dinlenme süresi ve mesai süresi içerisinde yemek molasına sahiptir.

5. Sonuç ve Tartışma

Arı kolonisi optimizasyon algoritması zor kombinasyonel problemler için oldukça efektif çözümler sunmaktadır. Şoför-hat-zaman çizelgeleme problemi oldukça fazla bileşen içeren ve çizelgelenmesi zor bir problemdir. Bu problemin çözümü için BCO algoritması önerilmiş ve sonuçlar önceki bölümde bildirilmiştir. Literatürde bu tür bir problemin çözüm kalitesinin farklı algoritmalar ile sınınamamış olması sebebiyle elde edilen sonuçlar hâli hazırda kullanılmakta olan çizelge ile kıyaslanmıştır. BCO’nun bu kıyaslama neticesinde daha az şoför ile çizelgelemeyi kabul edilebilir bir düzeyde bitirmiştir. Buna ek olarak seferler arası dinlenme süresi ve mesai saatleri içerisindeki yemek molası istasyon sorumlusu tarafından hazırlanan çizelgede her şoföre istisnasız verilememektedirken BCO ile hazırlanan çizelgede bu durum ortadan kaldırılmıştır.

Semboller

BCO: Arı Kolonisi Optimizasyonu

p : seçilebilecek sefere ait ihtimal değeri

d : iki sefer arasındaki zamansal uzaklık

t : iterasyon numarası

L : arının çözümüne olan sadakati

B : Yapay Arı

OFV : Amaç fonksiyonunun sayısal değeri

Teşekkür

Çalışmamıza yapıcı eleştirileri ile katkıda bulunan sayın hocalarımız Prof.Dr. Şirzat Kahramanlı, Dr. Oğuz Fındık Beyler'e ve mesai arkadaşlarımıza teşekkür ederiz.

Kaynaklar

- [1] Lucic P, Teodorovic D, Transportation Modeling: An Artificial Life Approach, Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence ICTAI'02, pp.216-223, 2002.
- [2] Wong LP, Low MYH, Chong CS, Bee Colony Optimization with Local Search for Traveling Salesman Problem, IEEE International Conference on Industrial Informatics (INDIN 2008), Dajeon,Korea, pp. 1019-1025, 2008.
- [3] Wong LP, Low MYH, Chong CS, A Bee Colony Optimization Algorithm for Traveling Salesman Problem, Second Asia International Conference on Modelling & Simulation, pp.818-823, 2008.
- [4] Nakrani S, Tovey C, On Honey Bees and Dynamic Server Allocation in Internet Hosting Centers, International Society for Adaptive Behavior 2004, 12(3-4): 223-240.
- [5] Teodorovic D, Dell'orco M, Bee colony optimization-A cooperative learning approach to complex transportation problems, In 10th EWGT Meeting and 16th Mini-EURO Conference, <http://www.iasi.cnr.it/ewgt/16conference/ID161.pdf>, 2005.

- [6] Chong CS, Low MYH, Sivakumar AI, Gay KH, A Bee Colony Optimization Algorithm to Job Shop Scheduling, Proceedings of the 2006 Winter Simulation Conference, pp.1954-1961, 2006.
- [7] Kıran MS, Arı Kolonisi ile Şoför-Hat-Zaman Optimizasyonu, Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, 2010.
- [8] Teodorovic D, Bee Colony Optimization (BCO), Innovations in Swarm Intelligence, Studies in Computational Intelligence 2009, 248:39-60, Springer.
- [9] Teodorovic D, Lucic P, Markovic G, Dell'Orco M, Bee Colony Optimization: Principles and Applications, 8th Seminar Neural Network Applications in Electrical Engineering NEUREL-2006, pp. 151-156, 2006.