DRONE DESTEKLİ KURYE ATAMA PROBLEMİNİN BENZETİM MODELLEMESİ VE OPTİMİZASYONU

ÖMER YILDIRIM - ALİ BARAN KAYAR - YAĞMUR GÖMEK

PROF. DR. ŞENER AKPINAR

Dokuz Eylül Üniversitesi Endüstri Mühendisliği Bölümü

ÖZET

Son yıllarda çevrim içi alışverişlere yönelik talep, özellikle COVID-19 pandemisiyle birlikte büyük artış göstermiştir. Bu durum, kargo lojistiğinde klasik olarak uygulanan havuz yöntemine ek olarak, müşteri taleplerine daha hızlı yanıt vermeyi hedefleyen tekli teslimat yöntemlerinin yaygınlaşmasına neden olmuştur. Kurye Atama Problemi, tekli teslimat sistemlerinde sıkça karşılaşılan ve Dinamik Rotalama Problemi ile Kapasiteli Araç Rotalama Problemi gibi çeşitli rotalama problemleriyle yakından ilişkili olan dinamik bir problem olarak ortaya çıkmaktadır. Bu çalışma, Kurye Atama Problemi'ni çözmek amacıyla drone destekli yeni bir sistem önermekte ve optimizasyon sürecinde daha hızlı aksiyon alabilmek için genetik algoritmadan yararlanmaktadır. Önerilen sistem, Python programlama dili ile kodlanmış, SimPy simülasyon kütüphanesi kullanılarak test edilmiş ve çeşitli verilerle test edilerek potansiyeli vurgulanmıştır.

GİRİŞ

Son yıllarda çevrim içi alışveriş talebindeki artış, e-ticaret şirketlerinin sayısında kayda değer bir yükselişe ve buna bağlı olarak rekabetin daha da yoğunlaşmasına neden olmuştur. Özellikle COVID-19 pandemisiyle birlikte birçok ülkede bireylerin evlerine kapanmak zorunda kalması, temel ihtiyaçların dahi internet üzerinden temin edilmesine olanak sağlayan yeni teslimat modellerinin önem kazanmasına yol açmıştır. Bu süreçte klasik havuz yöntemlerinin yanı sıra tekli teslimat yöntemleri de yaygınlaşarak kullanılmaya başlanmıştır.

Tekli teslimat yöntemi, hızı ön planda tutarak rekabet avantajı yakalamayı amaçlamaktadır. Bu süreçte Kurye Atama Problemi ve çeşitli stokastik süreçleri içeren Dinamik Rotalama Problemi gibi ilişkili problemlerle karşılaşılmasıyla birlikte sistemin durumuna bağlı olarak Kapasiteli Araç Rotalama Problemi ve benzeri rotalama problemleri de ortaya çıkabilmektedir. Bu durum, teslimat süreçlerinin optimize edilmesini zorlaştırmaktadır.

Projenin amacı, dinamik şartlar altında bir teslimat sistemini drone desteğiyle güçlendirerek teslimat sürecini daha verimli hale getirecek bir sistem geliştirmektir. Tasarlanan sistem, özellikle öncelikli siparişlerin söz konusu olduğu durumlarda sistemin yükünü hafifletmeyi ve siparişlerdeki genel gecikmeyi azaltmayı amaçlamaktadır. Daha hızlı teslimatlar, rekabet avantajını arttırabildiği gibi sistemin yapısı genel maliyetleri de düşürebilecek şekilde tasarlanmıştır.

Şekil 1 Sistemin Kuşbakışı Görüntüsü

KULLANILAN ARAÇLAR

python™ SimPy {OOP}

1. KURYE ATAMAYLA İLİŞKİLİ PROBLEMLER

Kurye Atama Problemi, dinamik süreçlerin dahil edilmesiyle Dinamik Rotalama Problemi ile beraber ele alınır. Kuryenin önce restorana uğrama zorunluluğu, Alma ve Teslim Etme Problemi başlığı altında değerlendirilir. Kapasite kısıtları, Kapasiteli Rotalama Problemi'nin ortaya çıkmasına neden olur. Birden çok kuryenin birden çok sipariş üstelenebilmesi Araç Rotalama Problemi'ni, en kısa yolu bulma hedefi ise En Kısa Yol Problemi'ni ortaya çıkartır. Sistemde birden fazla optimizasyon hedefinin olması ise Çok Amaçlı Optimizasyon Problemi'ne yol açar.

Senaryo	SSP	ARP	ATEP	DRP	EKYP	KRP	ÇAOP	KAP
1-1-1-1	✗	✗	✓	✓	✗	✓	✓	✗
1-1-1-N	✗	✗	✓	✓	✓	✓	✓	✗
1-1-N-1	✓	✗	✓	✓	✗	✓	✓	✗
1-1-N-N	✓	✗	✓	✓	✓	✓	✓	✗
1-N-1-1	✗	✗	✓	✓	✗	✓	✓	✗
1-N-1-N	✗	✗	✓	✓	✓	✓	✓	✗
1-N-N-1	✓	✗	✓	✓	✗	✓	✓	✗
1-N-N-N	✓	✗	✓	✓	✓	✓	✓	✗
N-1-1-1	✗	✗	✓	✓	✗	✓	✓	✓
N-1-1-N	✗	✗	✓	✓	✓	✓	✓	✓
N-1-N-1	✓	✓	✓	✓	✗	✓	✓	✓
N-1-N-N	✓	✓	✓	✓	✓	✓	✓	✓
N-N-1-1	✗	✗	✓	✓	✗	✓	✓	✓
N-N-1-N	✗	✗	✓	✓	✓	✓	✓	✓
N-N-N-1	✓	✓	✓	✓	✗	✓	✓	✓
N-N-N-N	✓	✓	✓	✓	✓	✓	✓	✓

Şekil 2 Önerilen Sistemin İlk Aşamasında Senaryolara Göre Karşılaşılan Problemler

Senaryo	SSP	ATEP	DRP	ÇAOP	KRP	DAP
1-1-1-1	✗	✓	✓	✓	✓	✗
1-1-N-1	✓	✓	✓	✓	✓	✗
N-1-1-1	✗	✓	✓	✓	✓	✓
N-1-N-1	✓	✓	✓	✓	✓	✓

Şekil 3 Önerilen Sistemin İkinci Aşamasında Senaryolara Göre Karşılaşılan Problemler

3. NESNE TABANLI PROGRAMLAMA

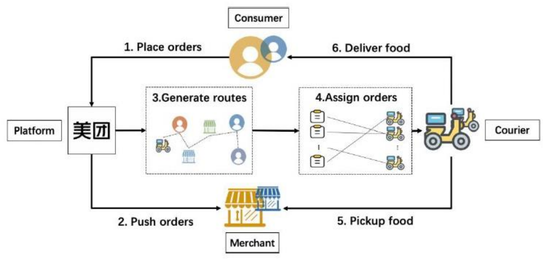
Nesne Tabanlı Programlama, varlıklar üzerindeki veri akışının denetimini kolaylaştırarak sistem modellemesinin oluşturulmasına katkı sağlar. Her bir varlığın kendine özgü niteliklerinin bulunması, sistemdeki varlık sayısına bağlı olmaksızın tek bir değişken ismiyle bu niteliklere ulaşılmasını sağlar.

Bu yöntemin sağladığı avantajlar sayesinde değişken kullanımı minimize edilirken program kodları kompakt bir şekilde düzenlenebilir. Sistemdeki her bir varlık (teslimat noktaları, kuryeler, drone'lar, restoranlar, depolar vb.) bu yöntemle nesne olarak tanımlanmıştır.

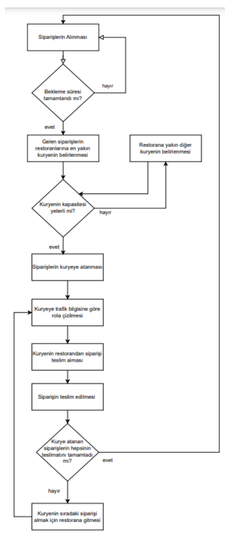
```python
class Order:
    def __init__(self, food, origin, destination, id, created_time, priority):
        self.food = food
        self.origin = origin
        self.destination = destination
        self.id = id
        self.name = f"order_{id}"
        self.prep_time = self.get_prep_time(food, origin)
        self.priority = priority

        if priority is True:
            self.expected_delivery_time = self.prep_time * P.prior_order_delivery_time_tolerance
        else:
            self.expected_delivery_time = self.prep_time * P.delivery_time_tolerance

        self.created_time = created_time
        self.delivered_time = None

    def get_prep_time(self, food, origin):
        df = pd.read_csv('data/order_details.csv')
        match = df[(df['food'] == food) & (df['restaurant id'] == origin)]
        return int(match.iloc[0]['prep time']) if not match.empty else 10
```

Şekil 7 Programda Sipariş Sınıfının Tanımlanması

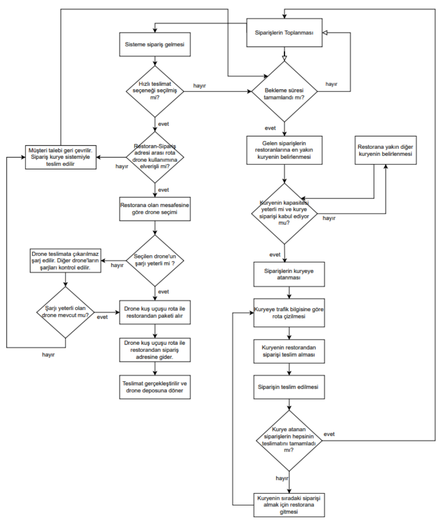
2. SİSTEMİN ALGORİTMİK TASARIMI



Şekil 4 Tekli Teslimat Yöntemine Dair Örnek Model



Şekil 5 Sistem Tasarımının Algoritmik Şeması



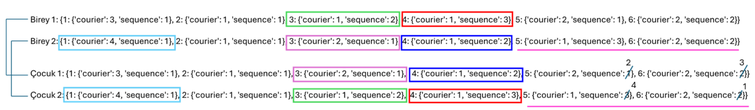
Şekil 6 Drone Destekli Hibrit Sistemin Algoritmik Şeması

5. GENETİK ALGORİTMA İLE OPTİMİZASYON

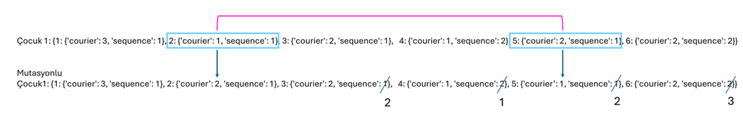
Başlangıç, ilgili restorana en yakın kuryenin seçilmesini içeren bir sezgisel yöntemle yapılır.

Çaprazlama işlemi temel olarak **siparişlerin hangi kuryeye atandığı bilgisini değiştirir. Siparişlerin sıra bilgileri de atamalarda gerçekleşen değişimler sonucunda güncellenir** ve tutarlı hale getirilir. Çözüm üzerindeki her sipariş için çaprazlama işleminin gerçekleştirilip gerçekleştirilmeyeceği önceden belirlenen bir olasılık (bu çalışmada %80) ile belirlenir. Çaprazlama sonucunda oluşan çocuklar (yeni çözümler) kurye kapasitelerinin aşılmaması açısından kontrol edilir.

Mutasyon sırasında çözümdeki siparişler arasında rastgele iki farklı sipariş seçilir. Seçilen siparişlerin atandığı kuryeler farklıysa bu iki siparişin kurye atamaları birbiriyle değiştirilir. Mutasyonun gerçekleşip gerçekleşmeyeceği de yine belirli bir mutasyon oranına (%20) göre rastlantısal olarak belirlenir. Tıpkı çaprazlamada olduğu gibi mutasyonda da sipariş sıralamaları yeniden düzenlenerek oluşabilecek tutarsızlıklar engellenir.



Şekil 10 Örnek Çözüm Üzerinde Çaprazlama



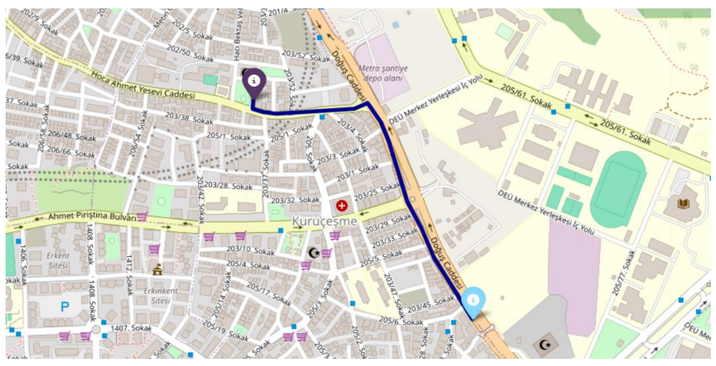
Şekil 11 Örnek Çözüm Üzerinde Mutasyon

4. API İLE PYTHON ENTEGRASYONU

```python
def get_route_duration(origin_location, destination_location):
    import requests
    import json

    url = "https://routes.googleapis.com/directions/v2:computeRoutes"
    payload = {
        "origin": {
            "location": {
                "latLng": {
                    "latitude": origin_location['loc_y'],
                    "longitude": origin_location['loc_x']
                }
            }
        },
        "destination": {
            "location": {
                "latLng": {
                    "latitude": destination_location['loc_y'],
                    "longitude": destination_location['loc_x']
                }
            }
        },
        "travelMode": "DRIVE",
        "routingPreference": "TRAFFIC_AWARE",
        "computeAlternativeRoutes": False,
        "languageCode": "tr",
        "units": "METRIC"
    }

    headers = {
        "Content-Type": "application/json",
        "X-Goog-Api-Key": API_KEY,
        "X-Goog-FieldMask": "routes.distanceMeters,routes.duration"
    }

    response = requests.post(url, headers=headers, data=json.dumps(payload))
```

Şekil 8 Google Routes API POST İsteği İçeriği

Python'da yer alan «Requests» kütüphanesi, POST isteklerinin atılmasına olanak sağlar. Araç trafiğine açık yollar, yolların anlık durumu ve mesafelerle ilgili verilerin alınması için Google Routes API, bu kütüphane ile istek atılarak kullanılmıştır. Dönen yanıt ise değişken olarak dışarı aktarılıp programda kullanılmıştır.



Şekil 9 Google Routes API Tarafından Döndürülen Rotanın Harita Üzerinde Görselleştirilmiş Hali

SONUÇ VE ÖNERİLER

```
[01:39:00] Atanmamış Siparişler: Sistemde atanmamış sipariş yok.
[01:39:12] courier_2 gecikmeli teslimat yaptı! Gecikme: 12.44 dakika
[01:39:18] courier_1 kuryesi restaurant_6 restoranına ulaştı. Sipariş hazırlanıyor.
[01:39:32] Yeni Sipariş -> ID: order_21, Lokasyon: delivery_adresi_68, Restoran: restaurant_4, Yiyecek: Tost, Beklenen Teslimat
Süresi: 8.16 dakika, Restoran - Adres Mesafesi: 3721.574218897097
[01:39:32] drone_3 dronuna order_21 siparişi atandı.
[01:40:20] drone_3 dronu restaurant_4 restoranına ulaştı. Sipariş hazırlanıyor.
[01:42:00] Atanmamış Siparişler: Sistemde atanmamış sipariş yok.
[01:44:16] Yeni Sipariş -> ID: order_22, Lokasyon: delivery_adresi_55, Restoran: restaurant_6, Yiyecek: Risotto, Beklenen
Teslimat Süresi: 27.20 dakika, Restoran - Adres Mesafesi: 2761.6315826425503
```

Şekil 12 Simülasyonun Çalışması Esnasında Sistem Kayıtları

Özellik	Değer
Toplam Teslim Edilen Sipariş Sayısı	374
Toplam Teslim Edilen Önemli Sipariş Sayısı	6
Toplam Geç Teslim Edilen Sipariş Sayısı	121
Toplam Geç Teslim Edilen Önemli Sipariş Sayısı	1
Toplam Gecikme Süresi (dk)	645,42
Toplam Kurye Kazası	0
Toplam Drone Kazası	0
Toplam Reddetme	29
Toplam Kat Edilen Mesafe (km)	2148,264339

Şekil 13 30 Saatlik Simülasyon Testinde Sistemin Performans Ölçütleri

ID	İsim	Toplam Kat Edilen Mesafe (km)	Toplam Kaza Sayısı	Toplam Reddetme Sayısı	Toplam Erken Teslim Zamanı (dk)	Toplam Geç Teslim Zamanı (dk)	Toplam Erken Teslim Edilen Sipariş Sayısı	Toplam Geç Teslim Edilen Sipariş Sayısı
1	courier_1	578,9093	0	4	512,25	189,72	63	32
2	courier_2	452,8499	0	8	514,87	119,21	60	25
3	courier_3	413,6398	0	5	407,27	143,41	47	21
4	courier_4	435,9649	0	7	371,66	122,99	51	24
5	courier_5	173,7702	0	2	167,6	39,01	20	11
6	courier_6	76,1252	0	3	72,64	27,63	7	7

Şekil 14 30 Saatlik Simülasyon Testinde Kuryelerin Performans Ölçütleri

Önerilen sistemin simülasyonu 30 saat boyunca çalıştırılmış ve elde edilen sonuçlar yan tarafta verilmiştir. Sonuçlar göstermektedir ki teslimat süresi beklentilerinin çok daha katı olduğu öncelikli siparişlerde bile drone'lar çok düşük oranda gecikmeye sebep olmaktadır. Diğer yandan teslimat süresi beklentilerinin daha gevşek olmasına karşılık moto kuryeler önemli miktarda sipariş geciktirebilmektedir.

Drone'ların hız açısından sağladığı avantaj barizdir. Bununla birlikte drone'lar operasyonel maliyetleri düşürebilme potansiyeline sahiptir. Sürdürülebilirlik açısından incelendiğinde ise drone'ların karbon emisyonlarının daha az olması, gürültü kirliliği ve trafik açısından olumlu sonuçlara sahip olması bu teknolojinin birçok açıdan faydalı olduğunu göstermektedir.

Olumlu özelliklerine karşılık sistemin tasarlanmasında varsayım olarak belirlenen maddeler sistemin gerçek hayat uygulaması için engel teşkil etmektedir. Bu projenin geliştirilmesi için drone'ların teslimat yöntemleri, enerji seviyelerinin yönetimi, yapay zekalarının gelişmişlik düzeyi ve stratejik konumları gibi konular üzerine yoğunlaşılması önerilmektedir.

KAYNAKÇA

Liang, Y., Luo, H., Duan, H., Li, D., Liao, H., Feng, J., … & Wang, L. (2024). Meituan's Real-Time Intelligent Dispatching Algorithms Build the World's Largest MinuteLevel Delivery Network. INFORMS Journal on Applied Analytics, 54(1), 84-101.