

Derin Pekiştirmeli Öğrenme ve Sezgisel Yöntemlerin Kapasite Kısıtlı Araç Rotalama Probleminde Entegre Kullanımı

A Hybrid Method by Integrating Deep Reinforcement Learning and Heuristics Approach for Capacitated Vehicle Routing Problem

Yasin Furkan Aktaş^{1,2}, Ahmet Murat Özbayoğlu¹

¹Bilgisayar Mühendisliği Bölümü, TOBB ETÜ, Ankara, Türkiye
{yaktas,mozbayoglu}@etu.edu.tr

²ASELSAN Araştırma Merkezi, ASELSAN, Ankara, Türkiye
{furkanaktas}@aselsan.com.tr

Özetçe —Çevrimiçi platformların yaygınlaşmasıyla insanlı/insansız yemek teslimatı, kargo teslimatı, hammadde teslimatı gibi problemler lojistik alanının önemini her geçen gün artırmaktadır. Lojistik alanındaki en önemli problemlerden birisi olan araç rotalama problemi ise kombinatorik bir problem olup problem uzayı büyüdükçe insan çabası ile çözümü bulmak uzun sürmekte ve hatta çoğu durumda çözümü bulmak mümkün olmamaktadır. Bu sebeple bu problemin çözümünün otonom olması önem arz etmektedir. Klasik sezgisel optimizasyon yöntemleri ile problemin çözümü mümkün olsa da problem boyutuna göre dakikalardan saatlere kadar uzun sürmekte ve bazen yeterince iyi çözüm verememektedir. Dikkat mekanizmalı derin pekiştirmeli öğrenme modelleri bu konuda büyük bir potansiyel barındırmaktadır. Fakat derin pekiştirmeli öğrenme metodlarının problem uzayı büyüdüğü durumlarda yeterli miktarda eğitilememesi durumunda metodlar iyi sonuç verse de optimal çözümünden zaman zaman uzaklaşabilmektedir. Bu çalışmada dikkat mekanizmalı derin pekiştirmeli öğrenme modelleri ile sezgisel yöntemleri hibrit bir şekilde kullanarak kabul edilebilir sürede daha iyi sonuçlar alınmaktadır.

Anahtar Kelimeler—Derin pekiştirmeli öğrenme, dikkat modelleri, sezgisel yöntemler, optimizasyon, araç rotalama problemi

Abstract—With the spread of online platforms, problems such as manned/unmanned food delivery, cargo delivery, raw material delivery, are increasing the importance of logistics day by day. Vehicle routing problem, which is one of the most important problems in the field of logistics, is a combinatorial problem and as the problem space grows, it takes a long time to find a solution with human effort and in most cases it is not even possible. Thus, it becomes essential for the solution of this problem to be autonomous. Although it is possible to solve the problem with classical heuristic optimization methods, it takes a long time and sometimes does not give a good enough solution. Deep reinforcement learning models with attention mechanisms have great potential in this regard. However, in case of insufficient training in large problem space, it is possible to get away from the optimal solution. In this study, better results are taken in an

acceptable time by using the deep reinforcement learning models with attention-model and heuristic methods in a hybrid way.

Keywords—Deep reinforcement learning, attention, heuristics, optimization, CVRP

I. GİRİŞ

Günümüzde çevrimiçi alışveriş ve tedarik platformlarının kullanımının yayılmasıyla tedarik zinciri ve lojistik konularının önemi ile doğru orantılı olarak araç rotalama problemi üzerine yapılan çalışmalar hızla artmaktadır. CVRP (Kapasite Kısıtlı Araç Rotalama problemi) NP-hard bir problem olmakla beraber tüm ihtimallerin denenerek çözülmesi neredeyse imkansız olmaktadır. Gurobi [1] ve CPLEX [2] gibi matematiksel kesin çözümleyiciler ile optimal çözümü bulmak [3]’de belirtildiği üzere 20 müşteri gibi küçük problem uzayında 15 dakika kadar uzun sürmekte ve problemin uzayının büyümesiyle birlikte 50,100 müşterili problem boyutlarında optimal çözümün bulunması neredeyse imkansız hale gelmektedir. VRP’nin ölçeklenebilirlik probleminin önüne geçmek için optimal çözüme yaklaşabilecek yöntemler öne sürülmüştür. VRP’nin daha basit versiyonu ve NP-Hard problem olan Gezin Satıcı Problemi (TSP) üzerine yapılan çalışmalar ile bulunan çözümler çoğu durumda optimal bulabilir ya da optimale yakınsayabilirken VRP’de problem uzayı ve kısıtların daha fazla olmasından dolayı optimal çözüme yaklaşma üzerine yapılan çalışmalar hala devam etmektedir.

Geçmişte yapılan çalışmalarda klasik sezgisel yöntemlerden olan genetik algoritma, karınca kolonileri algoritması gibi yöntemler uygulanmıştır. Fakat bu yöntemlerin probleme kabul edilebilir güzel bir sonuç vermesi üzere çok uzun sürmektedir. Örneğin [4]’de yapılan çalışmada 50,100,150 müşterili problem için sırasıyla 2,10,25 dakika boyunca optimizasyon ile sonuç bulunmuştur. Ardından TSP ile öne sürülen Lin

Kernighan sezgisel algoritması gibi rotalama problemlerine özgü sezgisel yöntemler ile daha iyi sonuçlar alındığı fakat hesaplama süresinin saatler mertebesinde uzun sürdüğü gözlemlenmiştir [3].

Sezgisel ve klasik optimizasyon yöntemleri ile problemi çözmek çok uzun sürdüğünden dolayı, CVRP'nin ölçeklenebilirlik problemini çözmek adına son yıllarda yapay öğrenme yöntemleri yaygınlaşmıştır. Denetimli öğrenme ile yeterince iyi öğrenen sistemler, problemin çözümünü hızlı bir şekilde bulabilir. Fakat eğitim için gerekli veriyi üretmek gerekmede ve yeterli miktarda optmale yakın çözüm üretmek kullanışlı bir yöntem olmamaktadır. Problem uzayını kümeleyerek her aracın rota çözümünü ayrı ayrı bulmak çok araçlı gezgin satıcı probleminde uygulanabilirken CVRP için bu yöntem uygun olmamaktadır. Bu yüzden son zamanlarda dikkat mekanizmalı derin pekiştirmeli öğrenme modelleri CVRP üzerinde yaygınlaşmıştır ve bu çalışmalarda bilinen en iyi çözümü üreten yöntemlere oldukça yakınsamış ve çözümleme süresini oldukça kısaltmıştır.

Bu çalışmada ise dikkat mekanizmalı derin pekiştirmeli öğrenme yöntemleri ve sezgisel yöntemler birbirine entegre edilerek iki katmanlı bir çözüm yöntemi oluşturulmuş ve kabul edilebilir hesaplama süresi ile daha iyi sonuçların elde edilmesi hedeflenmiştir.

Bildirinin kalan kısmı şu şekilde organize edilmiştir. 2. bölümde literatürde incelenen yöntemler özetlenmiş, 3. bölümde problem tanımı yapılmış, 4. bölümde deneylerin kurgusu ve uygulanış biçimi anlatılmış ve 5. bölümde deney sonuçlarının analiz ve değerlendirilmesi yapılmıştır. Son olarak 6. bölümde yapılan çalışma ile ilgili muhtemel gelecek çalışmalardan bahsedilerek 7. bölüm ile bildiri sonuçlandırılmıştır.

II. İLGİLİ ÇALIŞMALAR

A. Sezgisel ve Klasik Optimizasyon Yöntemleri

Sezgisel yöntemler ile CVRP ve varyantları olan rotalama problemleri üzerine uzun yıllar boyunca çalışılmıştır. Genel olarak birçok optimizasyon probleminde uygulanıp iyi sonuçlar verdiği gözlemlenen yerel arama algoritmaları [5], genetik algoritma [6], karınca kolonileri algoritmaları [4] bu problemler üzerinde de uygulanmıştır. Swap [7], 2-opt [8] gibi sezgisel yöntemler TSP probleminde iyi sonuçlar vermesine rağmen CVRP üzerinde uygulanmamıştır. Yine TSP probleminde uygulanarak ortaya çıkan ve genişletilen LKH3 adlı yöntem [9] Lin Kernighan Sezgiseli diye adlandırılan ve temel olarak 2-opt algoritmasının geliştirilmiş versiyonu olan yöntem TSP problemlerinde bilinen en iyi çözümleri bulabilmiş hatta bazılarında daha iyilerini bulduğu gözlemlenmiştir. LKH yöntemi genişletilmiş ve farklı araç rotalama problemlerine uyarlanmıştır. CVRP probleminde ise TSP de olduğu gibi en iyi çözümleri bulabilmektedir ve bundan dolayı [10],[3] gibi birçok çalışmalarda performans ölçütü bu yöntemle göre yapıldığı görülmektedir. Referans [11]'deki çalışmada ise problem uzayını küçültmek adına k-means, faktör analiz gibi yöntemler ile problemi alt problemlere kümeleyerek her araç için ayrı rotalama problemi çözümleme yöntemleri denenmiştir.

Daha büyük problem uzayını içeren araç rotalama problemlerinde alt seviye sezgisel yöntemler ile tabu arama [12]

yöntemi, yönlendirmeli yerel arama [13] yöntemleri gibi ön sezgisel yöntemler birleştirilerek çözümler üretilmiştir. Ön sezgisel yöntemler ve klasik optimizasyon yöntemlerinin kullanımı çok geniş olduğu için bu konuda yaygın kullanılan yazılım araçları da gelişmiştir. Concorde [14] aracı TSP için büyük ölçekte kesin sonuç verebilen ünlü bir yazılım kütüphanesi olduğu gibi, Gurobi [1] ve CPLEX ile [2] matematiksel olarak modellenen kombinatorik problemler için optimal çözümler bulunmakta, fakat problem uzayı büyüdükçe optimal çözüm bulunamamaktadır. OR-tools [15] Google tarafından geliştirilmiş ve araç rotalama konusunda kullanılan yaygın bir kütüphanedir. Bununla beraber ön sezgisel ve alt seviye sezgisel yöntemleri barındıran ve özelleştirilebilen bir kütüphanedir ve birçok CVRP çalışmalarında [3, 16] karşılaştırmalarda yer almaktadır.

B. Öğrenme Tabanlı Yöntemler

Sezgisel yöntemler ile yapılan çalışmaların ardından problemi daha ölçeklenebilir hale getirmek adına son yıllarda yapay öğrenme yöntemleri ile sıklıkla karşılaşılmaktadır. Yapay öğrenme yaklaşımında modele yeterli miktarda CVRP problemleri ve probleme uygun çözüm içeren veriseti ile besleyip eğiterek problemin test aşamasında çok hızlı bir şekilde kaliteli sonuçlar verebileceği düşünülmüştür [17]. Referans [18]'deki çalışmada öğrenme tabanlı bir yöntem kullanılmış ve kombinatorik optimizasyon problemleri üzerine daha başarılı olduğu gözlemlenen İşaretleyici Ağ Mimarisi (Pointer Network) sunulmuştur. Bu çalışmada önerilen mimari dikkat mekanizması ve yenilemeli sinir ağı (RNN) tabanlı kodlayıcı-kod çözücü (encoder-decoder) yapısına sahiptir. Denetimli öğrenme metotları kombinatorik problemlerinde kullanılması mümkün görünse de çok miktarda veriye ve zamana ihtiyaç duyulduğundan dolayı [3] çalışmasında işaretleyici mimarinin dikkat mekanizmasındaki RNN yapısını dönüştürücü (transformer) mimarilerindekine benzer olarak çok başlı dikkat (multi-head attention) ile düzenlemiştir. Ayrıca model parametrelerini çevre ortamı ile etkileşim sağlayarak denetimsiz öğrenme (reinforcement learning) yöntemini kullanmıştır. Bu çalışma ile eğitim süresini daha kısalttığı ve birçok sezgisel yöntemden daha iyi sonuç verdiği görülmüştür. [19] çalışmasında ise CVRP'nin farklı bir varyantı olarak heterojen araçlarda kapasite kısıtlı rotalama problemini derin pekiştirmeli öğrenme yöntemiyle çözmüş ve [3]'de bahsedilen mimariden esinlenmiştir. Heterojen araçların rotasını planladığı için bir kod çözücü yerine araç seçimi ve rota seçimini gerçekleştiren iki ayrı kod çözücü kullanılarak tek bir aksiyonda birleştirerek uygulanmıştır.

III. PROBLEM TANIMI

CVRP, Tablo I'de görüldüğü üzere temel olarak bir depo, N sayıda müşteri ve her bir müşteriden gelen değişken miktarda talepten oluşmaktadır. Problemin amacı, maksimum talep sınırı (c) olan minimum sayıda araçla toplam alınan yolu minimize etmek amacıyla en iyi çözümü bulmaktır yani Tablo I'deki amaç fonksiyonunu minimize etmektedir. Bir başka benzetimle elimizde tek araç tipinden oluşan araç filosu bulunmakta ve araçların alabileceği maksimum kargo hacmi c ile sınırlıdır. Her müşteriye bırakılacak kargo büyüklüğü ile talep miktarı eşleştirildiğinde minimum alınacak yol ve araç sayısı ile müşterilere kargo dağıtımı gerçekleştirilmesi hedeflenmektedir.

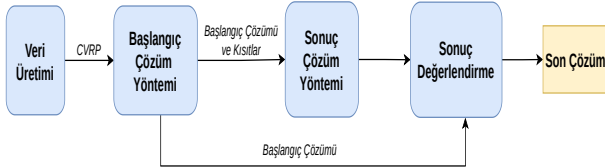
Değişken	Açıklama
N	Düğüm Sayısı
c	Araç Başı Maksimum Talep Kapasitesi
$m = (m_x, m_y)$	Depo Koordinatı
$X = \{x_1, x_2, \dots, x_i, \dots, x_N\}$	Düğüm Listesi
$x_i = (x_i, y_i)$	Düğüm Koordinatları
$P = \{p_1, p_2, \dots, p_i, \dots, p_N\}$	Talep Listesi
$p_i = (x_i, y_i)$	Düğümdeki Talep Miktarı
$V = \{v_1, v_2, \dots, v_i, \dots, v_k\}$	Araç Listesi
$v_i = (x_i, \dots, x_j, \dots, x_k)$	Araç Rotası
D_{v_i}	Araç rotasının toplam öklid mesafesi
$\text{Min} \sum_{v_i \in V} D_{v_i}$	Amaç fonksiyonu
$\sum_{x_j \in V_i} p_j \leq c$	Her Araç için Kapasite Kısıtı

Tablo I: Kapasite Sınırlı Araç Rotalama Problemi Değişkenleri

Temel olan CVRP probleminde araçlar depodan başlamakta ve yeterince araç sayısının filoda bulunduğu varsayılmaktadır.

IV. UYGULANAN YÖNTEM

Şekil 1’de görüldüğü üzere önerilen yöntem 4 ana modülden oluşmaktadır. İlk modül olan veri üretimi modülü, problemin verisini üretmek ve başlangıç çözüm yönteminin çalıştırabileceği uygun forma sokmaktan sorumludur. İkinci modül olan başlangıç çözüm yöntemi ise sonuç çözüm yönteminin iyi bir çözüme ulaşabilmesi için güzel bir başlangıç çözümü elde etmekten sorumludur. Üçüncü modül olan sonuç çözüm yöntemi ise başlangıç çözüm yönteminin ürettiği sonucu girdi olarak alıp iyileştirmekten sorumludur. Son modül olan sonuç değerlendirme modülü ise sonuç çözüm yönteminin kendisine tanımlanan hesaplama süresi limitinden önce elde ettiği çıktıyı ve başlangıç çözümünü karşılaştırıp daha iyi performans gösteren çözümü son çözüm olarak sunmaktadır.



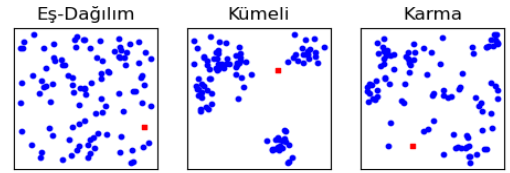
Şekil 1: Önerilen Yöntemin Akış Şeması

Referans [16],[17],[20]’deki çalışmalarda derin pekiştirmeli öğrenme yöntemleri kullanılarak CVRP için en iyi çözümlere yakın hızlı sonuçlar üretebildiği gösterilmiştir. Fakat bu çalışmalarda çok görülmese de derin pekiştirmeli öğrenme modelleri bazen yanlış davranışları öğrenmiş ya da yeterince keşfedemediği durumlarla karşılaşabilmektedir [21]. Diğer yandan sezgisel optimizasyon yöntemlerinde güzel bir başlangıç çözümü ile başlandığında problem çok daha hızlı bir şekilde optimele yakın çözüm vermektedir. LKH3 ve OR-tools CVRP’de en çok kullanılan sezgisel optimizasyon yöntemi olmakla beraber LKH3 hesaplama süresi uzun olmasına rağmen çoğu zaman en iyi sonuçları verebilmektedir. OR-tools ise LKH3’e göre daha kısa sürede sonuç vermesinin yanında [3] derin pekiştirmeli öğrenme ve LKH3ün performansının altında kalmaktadır. Sezgisel yöntemler ve DRL yöntemleri hibrit olarak

kullanıldığında sezgisel yöntemlerin yalın olarak kullanıldığı durumlara kıyasla problemi daha hızlı optimize edebildiği ve daha güvenilir bir şekilde çözüm elde etmesi sağlanmıştır. Böylelikle derin pekiştirmeli öğrenmenin hızlı çözüm üretmesi ve sezgisel optimizasyon algoritmalarının verilen çözümü iteratif olarak iyileştirme özelliği birbirini bütünlüştür.

V. DENEYLER

Yöntemi uygularken geçmişteki çalışmalarda en çok kullanıldığı gözlemlenen LKH3 [9], OR-tools [15] ve DRL(derin pekiştirmeli öğrenme) [3] yöntemleri baz alınmıştır. Baz alınan bu yöntemleri kombinasyonları ile hibritleştirerek hesaplama süresi ve sonuç iyileştirimi metriklerine göre değerlendirilmiştir. Referans [3]’deki çalışmada uygulanan dikkat (attention) mekanizması kullanılarak greedy, beam ve sample kod çözücü metotları ile karşılaştırmalar gerçekleştirilmiştir. Algoritmalar 10, 20, 50, 100 müşteri sayısı ile denenmiştir. Müşterilerin (demand) talep değerleri müşteri sayısı 10 olduğu durumda {5-50}, 20 olduğu durumda {3-33}, 50 olduğu durumda {2-25}, 100 olduğunda ise {2-20} aralığında rastgele dağılımlı değerler olarak belirlenmiştir. Depo ve müşteri konumları ise her deneyde rastgele atanarak seçilmektedir. Her aracın maksimum kapasitesi ise 100 olarak atanmıştır.



Şekil 2: Örnek CVRP verisetleri

Yöntemin müşteri lokasyonlarına bağımlı olup olmadığını test etmek için Şekil 2’de örnekendirildiği gibi tekdüze dağılımlı, kümelenmiş ve karma olmak üzere üç ayrı dağılımlı olduğu durumda testler yapılmıştır. Eşit dağılımlı olduğu durumda alana eşit ağırlıklarla, kümelenmiş dağılımda alan içinde 10-20-50-100 müşteri sayısında 3-3-4-5 rastgele küme merkezi seçerek kümelenmiş müşteri lokasyonları üretilir, karma dağılımda ise müşterilerin yarısı kümelenmiş diğer yarısı ise eşit rastgele dağılım ile seçilmektedir. OR-tools’un optimizasyon süresi için sırasıyla 10, 20, 50, 100 müşteri sayısı için 10, 12, 15, 20 olmak üzere kabul edilebilir süre limiti tanımlanmıştır. Deneyler Intel i7-10700 16 çekirdekli işlemci, 128 GB RAM ve Nvidia Quadro RTX4000 8 GB ekran kartına sahip bir platform üzerinde gerçekleştirilmiştir.

Deneyler gerçekleştirilirken LKH3 algoritmasının parametreleri max_trial=50000, move_type=5_special,runs=1 olarak seçilmiştir. Runs parametresi sonucu minimal düzeyde değiştirdiği gözlenmiş ve sonuçları daha hızlı elde etmek için 1 olarak atanmıştır. Kullandığımız DRL (AM/attention Model) modelleri ise [3] çalışmasından referans alınmıştır. DRL(AM)-Beam ve DRL(AM)-Sample için width=1280 parametreleri kullanılmıştır. CVRP problem kümeleri temel olarak [0-1] aralığında üretilmekte olduğu DRL(AM) modeller tarafından doğrudan kullanılabilir, fakat LKH3 ve OR-tools tamsayı maliyet ve kısıtlarla çalıştığı için kapasite kısıtları için 100, uzunluk hesapları için 1000 ile çarpılarak

Yöntem	N=10						N=20					
	Eşit		Kümelili		Karma		Eşit		Kümelili		Karma	
	Maliyet	Süre (sn)	Maliyet	Süre (sn)	Maliyet	Süre (sn)	Maliyet	Süre (sn)	Maliyet	Süre (sn)	Maliyet	Süre (sn)
LKH3	4392.1	3.531	4667.4	2.88	4957.2	3.86	5846.1	12.56	6302.2	11.15	6882.4	11.20
OR-tools	4234.0	10.01	4469.8	10.01	4932.1	10.01	5890.8	12.01	6274.9	12.01	6857.9	12.01
DRL(AM)-Greedy	4345.3	0.0187	4762.5	0.02	4906.6	0.014	5990.5	0.023	6499.8	0.03	6970.9	0.021
DRL(AM)-Beam	4189.6	0.0273	4447.2	0.022	4887.9	0.023	5821.9	0.04	6219.6	0.04	6810.9	0.04
DRL(AM)-Sampling	4276.0	0.0241	4575.2	0.018	4898.4	0.019	5921.1	0.031	6264.4	0.03	6847.2	0.03
LKH3-OR-tools	4234.0	13.531	4469.8	12.88	4932.1	13.86	5868.4	24.56	6271.7	23.15	6845.5	23.2
DRL(AM)-Greedy - OR-tools	4234.0	10.02	4469.8	10.02	4932.1	10.01	5871.5	12.02	6260.2	12.02	6857.9	12.03
DRL(AM)-Beam - OR-tools	4206.0	10.03	4469.8	10.02	4932.1	10.02	5828.0	12.04	6259.4	12.04	6858.5	12.04
DRL(AM)-Sample - OR-tools	4210.0	10.03	4457.4	10.02	4932.1	10.02	5871.5	12.03	6259.4	12.03	6857.9	12.03

Yöntem	N=50						N=100					
	Eşit		Kümelili		Karma		Eşit		Kümelili		Karma	
	Maliyet	Süre (sn)	Maliyet	Süre (sn)	Maliyet	Süre (sn)	Maliyet	Süre (sn)	Maliyet	Süre (sn)	Maliyet	Süre (sn)
LKH3	10941.8	66.04	8552.3	56.9	9983.5	52.75	15920.1	170.63	14154.2	159.8	15235.9	148.16
OR-tools	11479.1	15.01	8772.2	15.01	10262.2	15.01	16927.8	20.01	14724	20.01	16237.4	20.01
DRL(AM)-Greedy	11522.7	0.06	9271.7	0.05	10579.3	0.05	16862.9	0.09	15103.2	0.09	16241.7	0.1
DRL(AM)-Beam	11187.9	0.116	8771.4	0.13	10236.3	0.12	16413.0	0.33	14698.3	0.33	15818.8	0.34
DRL(AM)-Sampling	11213.1	0.118	8739.2	0.11	10248.0	0.10	16324.5	0.30	14611.6	0.3	15695.1	0.31
LKH3+OR-tools	10949.1	81.01	8533.0	71.93	10007.8	67.76	15876.0	190.63	13996.5	179.8	15205.4	168.2
DRL(AM)-Greedy +OR-tools	11147.0	15.06	8669.5	15.06	10118.1	15.05	16400.9	20.09	14348.8	20.1	15690.7	20.1
DRL(AM)-Beam + OR-tools	11059.0	15.118	8641.8	15.13	10047.5	15.12	16121.3	20.33	14334.6	20.33	15530.9	20.34
DRL(AM)-Sample + OR-tools	11092.5	15.12	8643.5	15.11	10079.1	15.11	16053.1	20.30	14245.7	20.30	15404.9	20.32

Tablo II: Performans ve Sonuç Karşılaştırmaları

dönüştürülmüştür. Son olarak deneylerde görülen maliyet değerleri bu dönüşümü baz alarak hesaplanmıştır.

VI. ANALİZ VE DEĞERLENDİRME

Tablo II'de listelenen gerçekleştirilmiş deneylerin sonuçlarına baktığımızda, oluşturulan verisetlerinin eş-dalımlı, kümelili, karma olarak karşılaştırdığımızda elde edilen çözümlerin maliyeti N=10,20 problem uzayında Karma>Kümelili>Eş Dalımlı olarak değiştiği N=50,100 problem uzayı için Eş Dalımlı > Karma > Kümelili olarak değiştiği görülmektedir. Bu sebeple N=10,20 gibi küçük boyutlu problemlerde problem için eş dalımlı olması avantaj iken N=50,100 gibi daha büyük boyutlu problemlerde dezavantajlı olduğu görülmektedir. Fakat, algoritmaların dalımlılarına göre maliyet değişimini incelediğimizde birbirine yakın bir artış azalma dalgası ile benzer davranış sergilediği görülmektedir. Böylelikle algoritmaların eş dalımlı, kümelili ve karma yapıları dalımlılarına dayanımlı olduğunu söyleyebiliriz.

Özellikle N= 10, 20 gibi küçük problemlerde derin pekiştirmeli öğrenme yöntemleri diğer sezgisel optimizasyon ve hibrit çözümlere göre yeterince iyi sonucu çok daha hızlı verebilmektedir. Diğer yandan derin pekiştirmeli öğrenme yöntemleri çok hızlı çözüm üretse de, [0.5-2.0] gün süren eğitim süreleri dezavantaj olarak değerlendirebilir. Yine de farklı dalımlı CVRP problemlerinde gösterdiği performans bu yöntemlerin potansiyelini kanıtlamaktadır.

Problem uzayının daha büyük olduğu durumlarda OR-tools'a başlangıç çözümü verilerek hibrit yöntemi uygulamanın maliyet değerini daha fazla düşürerek fayda sağladığını görmekteyiz. Bunun sebebi optimizasyon algoritmalarında büyük problem boyutlarında optimalite boşluğu (optimality gap) artmaktadır ve başlangıç çözümü ile çözüme sıfırdan değil çok daha önden başlamak performansı çok daha iyileştirmektedir. LKH3 algoritması ile hibrit kullanımın N=100 problem boyutunda iyileştirme sağlayarak en iyi sonuçları verse de

neredeyse tüm problem boyutları için optimalite boşluğu çok az olduğu için büyük bir iyileştirme sağlayamamıştır. Hibrit yaklaşım ile çözülen N=50,100 sonuçlarında DRL(AM) yöntemlerine ciddi iyileştirme sağlanmıştır çünkü N=50,100 için LKH3'e göre optimalite boşluğu daha geniştir. LKH3 N=50,100 ölçekteki problemlerde en iyi sonuç verse de diğer yöntemlere göre çok daha yavaş çalışmaktadır. Bir başka açıdan DRL(AM)-Sample+OR-tools yöntemi ile LKH3 yönteminin ürettiği çözüme çok yakın bir çözüme 148 saniyeden 20 saniye gibi daha kabul edilebilir bir sürede ulaşılabilir.

DRL(AM)-Greedy,DRL(AM)-Beam ve DRL(AM)-Sample, yöntemlerini incelediğimizde küçük boyutlu problemlerde çok iyi sonuçlar üretilmektedir. DRL(AM)-Beam modeli N=10,20 için daha iyi sonuç verirken, N=50 için DRL(AM)-Sample ile birbirine çok yakın sonuç vermektedir. N=100 için ise DRL(AM)-Sample diğer DRL(AM) yöntemlerine göre daha iyi sonuç vermektedir.Bunun sebebi problem boyutu büyüdüğünde beam kod çözümü yapısında sıradaki çıktıyı seçerken arama ağacının sınırı çözümü sınırlamaktadır. Arama ağacının maksimum genişliği 1280 olarak sabit olduğu için problem büyüdüğünde sonuç performansı sample yöntemine göre bir nebze daha düşmüştür. Genişlik değerinin daha büyük bir değer verilmesi durumda ise daha yavaş çalışma gibi dezavantajı bulunmaktadır.

Sonuç olarak özetle, DRL(AM) yöntemi küçük boyutlu (N=10,20) CVRP problemleri için iyi sonuçlar vermekte ve hibrit yöntemin faydası görülemiyor olsa da daha büyük boyutlu (N=50,100) CVRP problemleri için sonuçtaki iyileştirme bariz bir şekilde görülmektedir.

VII. GELECEK ÇALIŞMALAR

Bu çalışmada kapasite kısıtlı araç rotalama problemi ele alınmıştır. Gelecek çalışmalarda araç kapasitelerinin farklı olduğu, heterojen hızlara sahip olan problemler ele alınabilir. Ayrıca deneylerden elde edilen sonuçlar ışığında prob-

leme göre hangi yöntemin daha uygun olduğunu seçen akıllı mekanizmalar geliştirilebilir.

Araç rotalama probleminde sadece depodan müşteriye kargo alımı değil, müşteriden al- müşteriye/depoya bırak problemleri ile birleştirilebilir.

VIII. SONUÇ

Yapılan çalışma kombinatorik ve NP-hard bir problem olan kapasite kısıtlı araç rotalama problemi(CVRP) üzerine gerçekleştirilmiştir. Bu çalışma kapsamında öncelikle II’de özetlenen literatür araştırması yapılmıştır. Literatürde incelenen yöntemlere bakıldığında karınca kolonileri veya genetik algoritmalar gibi klasik sezgisel optimizasyon yöntemleri problemi çözmek için 5-20 dk aralığında çözümleme süresine ihtiyaç duymaktadır. LKH3 gibi rota planlama problemine özgü sezgisel optimizasyon yöntemleri ise çok iyi çözümler üretmesine rağmen hesaplama süresi saatlerce sürebilmektedir. Güncel yayınlarda yaygın olarak çalışılan dikkat mekanizmalı derin pekiştirmeli öğrenme yöntemi ise LKH3 kadar iyi bir performans göstermese de OR-tools’dan daha iyi bir çözümü literatürdeki diğer yöntemlerden daha hızlı bir şekilde elde edebilmektedir. Fakat, derin pekiştirmeli öğrenme yöntemlerinin eğitimi günlerce sürebilmektedir ve problemi yeterince öğrenme garantisi bulunmamaktadır. Bu yüzden daha emniyetli olduğu düşünülen ve IV’de belirtilen iki katmanlı hibrit bir çözüm yöntemi önerilmiştir.

Önerilen hibrit yöntemi test edip literatürdeki yöntemler ile karşılaştırmak amacıyla N=10,20,50,100 problem boyutlarında ve problem kümesinin eş dağılımlı, kümeli ve karma olduğu konfigürasyonlarda deney düzenekleri oluşturulmuştur. DRL, LKH3, OR-tools gibi yöntemler yalın olarak ve DRL+OR-tools, LKH3+OR-tools gibi hibrit olarak çalışan yöntemler oluşturulan deney düzeneklerinde test edilmiş ve V’de sunulmuştur. Elde edilen sonuçlar ışığında küçük problem uzayında (N=10,20) yalın yöntemler yeterince iyi sonuç verdiği için hibrit yöntemler ile pek iyileştirme görünmese de, geniş CVRP uzayındaki (N=50,100) sonuçlarda zaman ve sonuç açısından iyileştirme sağladığı görülmüştür.

ALINTILAR

- [1] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2022. URL: <https://www.gurobi.com>.
- [2] IBM ILOG Cplex. “V12. 1: User’s Manual for CPLEX”. In: *International Business Machines Corporation* 46.53 (2009), p. 157.
- [3] Wouter Kool, Herke van Hoof, and Max Welling. *Attention, Learn to Solve Routing Problems!* 2018.
- [4] John E. Bell and Patrick R. McMullen. “Ant colony optimization techniques for the vehicle routing problem”. In: *Advanced Engineering Informatics* 18.1 (2004), pp. 41–48.
- [5] Emile Aarts and Jan K. Lenstra. *Local Search in Combinatorial Optimization*. 1st. USA: John Wiley & Sons, Inc., 1997. ISBN: 0471948225.
- [6] Barrie M. Baker and M.A. Ayeche. “A genetic algorithm for the vehicle routing problem”. In: *Computers & Operations Research* 30.5 (2003), pp. 787–800.

- [7] Cathy Wu et al. “Optimizing the diamond lane: A more tractable carpool problem and algorithms”. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. 2016, pp. 1389–1396.
- [8] G. A. Croes. “A Method for Solving Traveling-Salesman Problems”. In: *Operations Research* 6 (1958), pp. 791–812.
- [9] S. Lin and B. W. Kernighan. “An Effective Heuristic Algorithm for the Traveling-Salesman Problem”. In: 21.2 (Apr. 1973), pp. 498–516.
- [10] Sirui Li, Zhongxia Yan, and Cathy Wu. *Learning to Delegate for Large-scale Vehicle Routing*. 2021.
- [11] Basma Hamdan, Hamdi Bashir, and Ali Cheaitou. “A novel clustering method for breaking down the symmetric multiple traveling salesman problem”. In: *Journal of Industrial Engineering and Management* 14.2 (2021).
- [12] Gulay Barbarosoglu and Demet Ozgur. “A tabu search algorithm for the vehicle routing problem”. In: *Computers & Operations Research* 26.3 (1999), pp. 255–270.
- [13] Philip Kilby, Patrick Prosser, and Paul Shaw. “Guided Local Search for the Vehicle Routing Problem with Time Windows”. In: *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Ed. by Stefan Voß et al. Boston, MA: Springer US, 1999, pp. 473–486.
- [14] Concorde- A TSP Solver. URL: <https://www.math.uwaterloo.ca/tsp/concorde.html>.
- [15] Laurent Perron and Vincent Furnon. *OR-Tools*. Version 7.2. Google. URL: <https://developers.google.com/optimization/>.
- [16] Mohammadreza Nazari et al. *Reinforcement Learning for Solving the Vehicle Routing Problem*. 2018.
- [17] Hanjun Dai et al. *Learning Combinatorial Optimization Algorithms over Graphs*. 2017.
- [18] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. *Pointer Networks*. 2015.
- [19] Jingwen Li et al. “Deep Reinforcement Learning for Solving the Heterogeneous Capacitated Vehicle Routing Problem”. In: *IEEE Transactions on Cybernetics* (2021), pp. 1–14.
- [20] Irwan Bello et al. *Neural Combinatorial Optimization with Reinforcement Learning*. 2016.
- [21] Alex Irpan. *Deep Reinforcement Learning Doesn’t Work Yet*.