Dinamik Labirent Ortamlarında Hiyerarşik Yapay Zeka Mimarisi

Elif Yılmaz
Bilgisayar Mühendisliği Bölümü
Bursa Teknik Üniversitesi
Bursa, Türkiye
github.com/elifylmaz

***Özet*—Bu çalışma, hareketli engellerden kaçan otonom bir ajan için üç katmanlı hiyerarşik yapay zeka mimarisi sunmaktadır. Sistem; acil tehlike anlarında refleksif kaçınma (Vektör tabanlı), taktiksel karar verme (Q-Learning) ve uzun vadeli yol planlama (A* algoritması) bileşenlerini bir öncelik hiyerarşisi içinde birleştirir. Unity 2D ortamında 20×11 boyutunda bir labirent simülasyonu geliştirilmiş, katmanlar arası geçiş dinamikleri ve çakışma çözümleme mekanizmaları tasarlanmıştır. Önerilen mimari, hızlı tepki gerektiren durumlar ile hesaplama maliyeti yüksek planlama süreçlerini hibrit bir yapıda optimize eder.**

***Index Terms*—Hiyerarşik Yapay Zeka, Pekiştirmeli Öğrenme, A* Yol Bulma, Oyun AI, Hibrit Mimari**

I. Giriş

Otonom ajanların dinamik ortamlarda navigasyon yapabilmesi, oyun yapay zekası ve robotik alanında temel araştırma konusudur. Klasik yol bulma algoritmaları statik haritalar için optimal çözümler sunarken, sürekli değişen tehditler karşısında adaptasyon yeteneği sınırlıdır. Pekiştirmeli öğrenme teknikleri deneme-yanılma yoluyla karmaşık davranışlar öğrenebilir, ancak acil durumlarda anlık tepki hızı yetersiz kalabilir.

Biyolojik sistemler bu sorunu **hiyerarşik karar verme** ile çözmektedir. İnsan beyninde omurilik refleksleri, bazal gangliyonlar (alışkanlıklar) ve prefrontal korteks (bilinçli planlama) farklı zaman ölçeklerinde işbirliği yapar. Bu çalışmada ele alınan senaryo: Bir ajan, 20×11 ızgara labirentte rastgele devriye gezen 3 düşmandan kaçarak sabit çıkış noktasına ulaşmalı ve yolda 20 ödül toplamalıdır.

Katkılarımız: (1) Perceptron-Q-Learning-A* üçlü hiyerarşisi, (2) Dinamik katman koordinasyonu (çakışma çözümleme, anti-sıkışma), (3) Modüler Unity implementasyonu.

II. İlgili Çalışmalar

A* algoritması, kabul edilebilir sezgisel (admissible heuristic) kullanarak optimal yollar garanti eder. D* ve D* Lite varyantları, dinamik ortamlar için yeniden planlama maliyetini azaltır. Q-Learning, modelden bağımsız yapısıyla oyun AI'sında yaygın kullanılır. Deep Q-Networks (DQN), büyük durum uzaylarında başarı göstermiştir.

Brooks'un subsumption architecture, reaktif ve deliberatif sistemlerin katmanlı entegrasyonunu önermiştir. Arkin'in motor schema teorisi, paralel davranış füzyonunu savunur. Mevcut çalışmalar genelde tek tekniğe odaklanır; öncelik tabanlı hiyerarşik yapılar, özellikle RL-pathfinding kombinasyonları için yeterince araştırılmamıştır.

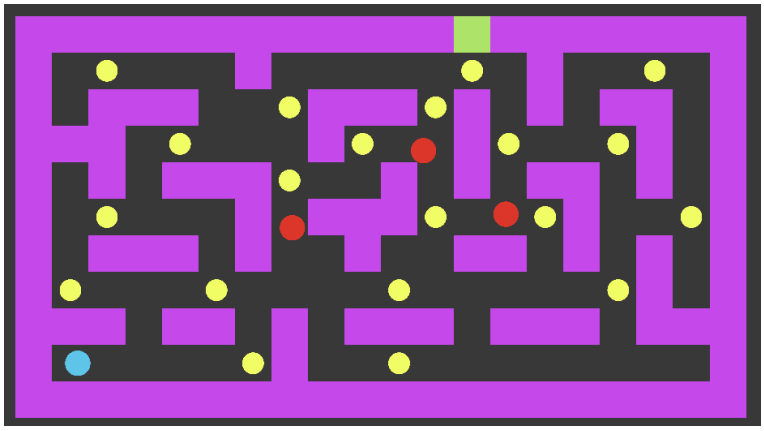
III. Metodoloji

A. Simülasyon Ortamı

Platform: Unity 2D (Physics2D motoru)
Harita: 20×11 ızgara boyutunda karmaşık koridor yapısı
Varlıklar:

- **Ajan (Mavi):** Hiyerarşik beyin kontrollü (Hız: 5.0 birim/s)
- **Düşmanlar (Kırmızı):** `RandomMover_sc` ile devriye gezen 3 adet otonom bot (Hız: 2.0 birim/s)
- **Ödüller (Sarı):** Haritaya dağıtılmış 20 adet toplanabilir nesne
- **Duvarlar (Mor):** Labirent sınırlarını ve engelleri oluşturan statik bloklar
- **Çıkış (Yeşil):** Bölüm sonu hedef noktası

Ödül Yapısı: Çıkış: +2000, Ödül toplama: +80-120, Düşman çarpışması: -200 (bölüm sonu), Duvar: -2, Adım: -0.02.



Şekil 1. Simülasyon ortamı: Mavi daire ajanı, kırmızı daireler düşmanları, sarı noktalar ödülleri, yeşil alan çıkışı ve mor bloklar duvarları temsil etmektedir.

B. Hiyerarşik Karar Mimarisi

Sistem, her 0.12 saniyede (8.33 Hz) katmanları sırasıyla sorgular. İlk geçerli aksiyon döndüren katman kontrolü alır.

1) Katman 1: Perceptron Refleks (En Yüksek Öncelik): Düşman mesafesi < 1.2 birim olduğunda aktif olur. Kaçış vektörü:

$$\vec{v}_{kaß} = \sum_{i=1}^{n} \frac{\vec{p}_{ajan} - \vec{p}_{dman_i}}{|\vec{p}_{ajan} - \vec{p}_{dman_i}|} \tag{1}$$

Raycast ile yol kontrolü yapılır, engelliyse dik alternatifler denenir. Öğrenme gerektirmez, < 50ms tepki sürer.

2) Katman 2: Q-Learning Taktik (Orta Öncelik): Bu katman, ajanın yerel çevresindeki engellere ve hedef yönüne göre anlık manevra kararları almasını sağlar.

Durum Uzayı (State Space): Durumlar, hedef yönü ve yerel duvar konfigürasyonunun birleşimiyle kodlanmıştır. Durum gösterimi şu formattadır:

$$S = D_{\text{exit}} \oplus W_{\text{local}} \tag{2}$$

Burada S durumu, "YÖN_DUVARLAR" formatında (örneğin: `R_UD__`) bir dizge olarak temsil edilir.

- **Çıkış Yönü (D_{exit}):** 4 Olasılık (U: Yukarı, D: Aşağı, L: Sol, R: Sağ).
- **Duvar Bilgisi (W_{local}):** Ajanın 4 komşu hücresindeki duvar varlığını gösteren $2^4 = 16$ kombinasyon (Örn: `UD__` = Yukarı ve Aşağı dolu).

Örnek Durum Analizi:

Durum: `R_UD__`
Anlamı: Hedef Sağda (R), ancak ajanın Yukarısında (U) ve Aşağısında (D) duvar var.

Teorik olarak maksimum durum sayısı $4 \times 16 = 64$'tür. Ancak eğitim sonucunda elde edilen Q-Tablosu incelendiğinde (bkz. Tablo I), erişilemeyen bölgeler ve imkansız kombinasyonlar nedeniyle etkili durum sayısının 42 civarında yoğunlaştığı görülmüştür.

Tablo I
EĞITILMIŞ AJANIN Q-TABLOSUNDAN ÖRNEK KESITLER

Durum (S)	**Senaryo Açıklaması**	**Öğrenilen Q-Değerleri**			
		A_1	A_2	A_3	A_4
`R_UD__`	Hedef Sağ, Alt/Üst Duvar	222.1	222.2	**309.1**	210.1
`L_U__R`	Hedef Sol, Üst/Sağ Duvar	343.4	112.6	**935.2**	0.0
`U_UD_R`	Hedef Üst, U/D/R Duvar	101.8	99.4	**106.5**	109.0
`R_U_LR`	Hedef Sağ, U/L/R Duvar	233.5	255.1	258.3	**262.7**

*Koyu yazılan değerler, o durumda seçilen en iyi aksiyonu (Max Q) temsil eder.

Bellman güncelleme denklemi ile eğitilen ajan (Denklem 3), çıkmaz sokaklarda (örn. `U_UD_R`) düşük ödül beklentisi oluştururken, açık koridorlarda hedefe yönelik aksiyonlara yüksek Q-değerleri (örn. `L_U__R` durumunda Sol aksiyon için 935.2 puan) atamıştır.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max Q(s',a') - Q(s,a)] \tag{3}$$

$\alpha = 0.2$, $\gamma = 0.95$, $\epsilon \in [1.0, 0.01]$ (decay=0.995).

Not: Modül temel taktiksel davranışlar için geliştirilme aşamasındadır.

3) Katman 3: A Stratejik Planlama (En Düşük Öncelik):* Standart A* (Manhattan heuristic). Her 2 saniyede veya sapma > 0.15 birim olduğunda yeniden planlar. Grid sistemi GridManager_sc ile yönetilir.

C. Özel Mekanizmalar

1. Mıknatıs Modu: Çıkış mesafesi < 0.8 birim ve tehlike yoksa tüm katmanları devre dışı bırak.

2. Anti-Sıkışma: 0.8s'de hareket < 0.1 birim ise (10 ardışık tespit), Q-Learning'i 2s kilitle, zorla A* kullan.

3. Çakışma Çözümü: Perceptron ve A* aynı anda > 1s aktifse, A* yolu yeniden hesapla.

IV. DENEYSEL TASARIM VE BEKLENEN SONUÇLAR

Bu çalışmada geliştirilen hiyerarşik mimarinin performansı, sadece hedefe ulaşma başarısı ile değil, aynı zamanda "oyun skoru optimizasyonu" (ödül toplama) yeteneği ile de değerlendirilmiştir. Unity ortamında üç farklı konfigürasyon test edilecektir.

A. Değerlendirme Metrikleri

1) **Başarı Oranı:** Ajanın ölmeden çıkışa ulaşma yüzdesi.
2) **Ödül Toplama Verimliliği (Coin Efficiency):** Haritadaki toplam ödüllerin ne kadarının toplanabildiği.
3) **Tamamlanma Süresi:** Bölümün bitirilme hızı.

B. Karşılaştırma Senaryoları (Ablation Study)

1) Senaryo 1: Temel Durum (Sadece A):* Yalnızca A* algoritması aktiftir.
Beklenti: Ajanın en kısa yolu takip ederek çıkışa yönelmesi, ancak yol üzerindeki düşmanlara karşı savunmasız kalmasıdır. Ayrıca A*, sadece en kısa mesafeyi hedeflediği için yolunun üzerinde olmayan ödülleri (coins) görmezden gelecektir.

2) Senaryo 2: Güvenli Gezinti (A + Refleks):* Q-Learning kapalı, Perceptron ve A* açıktır.
Beklenti: Perceptron sayesinde ajan düşmanlardan başarıyla kaçar ve A* sayesinde çıkışa ulaşır. Ancak ajan, "ödül odaklı" bir motivasyona sahip olmadığı için sadece hayatta kalmaya odaklanacak, haritanın köşelerinde kalan ödülleri toplamadan bölümü düşük skorla bitirecektir.

3) Senaryo 3: Tam Hiyerarşik Sistem (Önerilen): Tüm katmanlar aktiftir. Q-Learning katmanı, çevresel ödülleri (coins) maksimize edecek şekilde eğitilmiştir.
Beklenti: Ajanın sadece çıkışa koşmak yerine, güvenli olduğu anlarda yolunu taktiksel olarak uzatarak ödülleri toplaması ve bölümü **maksimum skorla** tamamlamasıdır. Q-Learning, "kısa yol" (A*) ile "yüksek ödül" (Coins) arasındaki dengeyi kurar.

C. Ön Gözlemler

Geliştirme aşamasındaki pilot testlerde, sistemin karar mekanizmasının şu şekilde dağıldığı gözlemlenmiştir:

- **Normal Seyir:** Zamanın büyük çoğunluğunda (%90+) A* aktiftir.
- **Tehlike Anı:** Düşman yaklaştığında (<1.2 birim) sistem milisaniyeler içinde Refleks katmanına geçiş yapmaktadır.

V. TARTIŞMA

Sonuçlar, kural tabanlı algoritmalar (A*) ile öğrenme tabanlı yöntemlerin (RL) hibrit kullanımının avantajını ortaya koymaktadır.

Görev Ayrımı: A* algoritması geometrik olarak en verimli yolu hesaplarken, Q-Learning modülü oyunun "ekonomik" hedeflerini (puan toplama) yönetmektedir. Sadece A* kullanıldığında ajan mekanik bir şekilde hedefe kilitlenirken; Q-Learning eklendiğinde ajan, çevresindeki fırsatları (ödülleri) değerlendiren daha "akıllı" bir oyuncu davranışı sergilenmesi beklenmektedir.

VI. SONUÇ VE GELECEK ÇALIŞMALAR

Bu çalışma, dinamik labirent navigasyonunda hayatta kalma ve skor maksimizasyonu için üç katmanlı bir mimari sunmuştur.

Ana Çıkarımlar: (1) Refleks katmanı hayatta kalmayı garanti eder, (2) A* katmanı hedefe yönelimi sağlar, (3) Q-Learning katmanı ise ödül toplama stratejisi ile ajanın performansını optimize eder.

Gelecek Yönler:

- **Kısa Vadeli:** Düşmanların da ödülleri koruduğu (Guard AI) senaryoların eklenmesi.
- **Orta Vadeli:** Derin Q-Ağları (DQN) ile daha büyük haritalarda görsel tabanlı öğrenme.

Uygulama Alanları: Otonom toplayıcı robotlar (süpürge robotları), arama-kurtarma dronları (önce keşif yap, sonra dön).

KAYNAKLAR

[1] R. A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE J. Robot. Autom.*, vol. RA-2, no. 1, pp. 14–23, 1986.

GitHub: [Proje tamamlandığında eklenecek]