Genel Ağırlık Atamaları için Verimli Bir Haydut Algoritması
An Efficient Bandit Algorithm for General Weight Assignments

Kaan Gökçesu[1], Tolga Ergen[1], Selami Çiftçi[2] ve Süleyman S. Kozat[1]
[1]Elektrik ve Elektronik Mühendisliği Bölümü, İhsan Doğramacı Bilkent Üniversitesi, Ankara, Türkiye
{gokcesu,ergen,kozat}@ee.bilkent.edu.tr
[2]Turk Telekom Labs, İstanbul, Türkiye
selami.ciftci@turktelekom.com.tr

***Özetçe* —Bu bildiride, muhalif çok kollu haydut problemi çalışılmış ve genel olarak uygulanabilen verimli bir haydut kolu seçimi yapısı sunulmuştur. Haydut kollarının kayıpları üzerinde hiçbir istatistiksel varsayım yapılmadığı için, bu bildirideki sonuçlar bireysel diziler şeklinde geçerli olmayı garanti etmektedir. Önerilen yapı haydut kolu seçim dizileri üzerinde genel ağırlık atamalarını kullanarak en iyi pişmanlık sınırlarını elde etmektedir. Bu yüzden, bu yapı çok sayıda uygulamada kullanılabilir.**

Anahtar Kelimeler—muhalif çok kollu haydut, genel yapı, değiştirme haydutu, verimli uygulama.

***Abstract*—In this paper, we study the adversarial multi armed bandit problem and present a generally implementable efficient bandit arm selection structure. Since we do not have any statistical assumptions on the bandit arm losses, the results in the paper are guaranteed to hold in an individual sequence manner. The introduced framework is able to achieve the optimal regret bounds by employing general weight assignments on bandit arm selection sequences. Hence, this framework can be used for a wide range of applications.**

Keywords—adversarial multi-armed bandit, general framework, switching bandit, efficient implementation.

I. Giriş

Son dönemlerde, çok kollu haydut yaklaşımları birçok gerçek hayat uygulamasında kullanılabildikleri için önemli biçimde çalışılmaktadır [1], [2]. Muhalif çok kollu haydut problemlerinde [3], M tane haydut kolu bulunmakta ve her bir raunt t'de, olasılıksal olarak bir kol seçilmektedir. Çevrimiçi seçim olan $\{u_t\}_{t\geq 1}$, $u_t \in \{1, 2, \dots, M\}$ verisine dayanarak, sadece seçilen kolun kaybı olan $\{l_{t,u_t}\}_{t\geq 1}$, $l_{t,u_t} \in [0, 1]$ alınmaktadır. Notasyon kolaylığı için $l_{t,u_t} \in [0, 1]$ varsayılmaktadır, ancak, bildirideki çıkarımlar kaydırma ve seviyelendirmeden sonra herhangi bir sınırlı kayıp için geçerlidir. T raunt bir oyunda, $\mathbf{u}_T = [u_1, \dots, u_T]^T$. sütun vektörü kullanıcının T zamanına kadar yaptığı seçimleri göstermektedir. $\mathbf{s}_T = [s_1, \dots, s_T]^T$ sütun vektörü rastgele olmayan T uzunluğundaki haydut kol seçim dizisini göstermektedir. öyleki her bir t için $s_t \in \{1, 2, \dots, M\}$ geçerlidir. Bildirinin geri kalan kısmında, $\mathbf{s}_T$ gibi haydut kol seçim dizileri bir strateji olarak belirtilmektedir. $\boldsymbol{l}_{\mathbf{s}_T} = [l_{1,s_1}, \dots, l_{T,s_T}]^T$ ise $\mathbf{s}_T$ stratejisinin kayıp dizisini göstermektedir. Böylece, $\mathbf{u}_T$ dizisinin kaybı $\boldsymbol{l}_{\mathbf{u}_T} = [l_{1,u_1}, \dots, l_{T,u_T}]^T$ olmaktadır. Burada, haydut kollarının davranışları üzerinde herhangi bir istatistiksel model varsayılmayan bir muhalif haydut ortamında çalışılmaktadır [4] ve önerilen algoritmalar bireysel dizi şeklinde çalışmayı garanti etmektedir. Algoritmanın t anındaki çıkışı olan u_t kesin bir biçimde çevrimiçi ve rastgeledir, ve çıkış sadece aşağıdaki gibi geçmişteki seçimler ve gözlemlenmiş kayıpların fonksiyonudur:

$$u_t \triangleq u_t(\boldsymbol{l}_{\mathbf{u}_{t-1}}; \mathbf{u}_{t-1}), \;\; u_t \in \{1, ..., M\}. \tag{1}$$

Herhangi bir $\mathbf{s}_T$ stratejisinin birikimli kayıp fonksiyonu $L_{\mathbf{s}_T} = \sum_{t=1}^{T} l_{t,s_t}$ olarak gösterilmektedir. Kayıp dizisi üzerinde bir varsayım yapılmadığından, en iyi strateji $\mathbf{s}^*_T = [s^*_1, \dots, s^*_T]$'ye göre performans aşağıdaki gibi tanımlanmaktadır:

$$\mathbf{s}^*_T = \arg\min_{\mathbf{s}_T} L_{\mathbf{s}_T} \quad \text{veya} \quad s^*_t = \arg\min_{s_t} l_{t,s_t}, 1 \leq t \leq T. \tag{2}$$

Performansı tanımlamak için pişmanlık kavramı şu şekildedir:

$$R_T \triangleq \sum_{t=1}^{T} l_{t,u_t} - \sum_{t=1}^{T} l_{t,s^*_t} = L_{\mathbf{u}_T} - L_{\mathbf{s}^*_T} \tag{3}$$

T anına kadar birikmiş pişmanlıktır.

Herhangi bir kayıp dizisi üzerinden en iyi stratejinin performansını elde etmek için M^T stratejinin her biri deterministik uzman olarak düşünülüp, üstel performans ağırlıkları ile birleştirilebilir [5]. Ancak, karışım algoritmaları $O(\sqrt{T \log N})$ pişmanlığı ve $O(N)$ hesaplama karmaşıklığına sahip olduğu için bu üstel sayıda algoritmaların basit bir birleşimi, üstel zamanda yok olmayan bir pişmanlık sınırı $O(T)$ üretmektedir [5]. Bu yüzden, çokterimli zaman içinde kaybolan bir pişmanlığı çevrimiçi olarak elde etmek için stratejilerin akıllıca birleştirilmesi ve ağırlıkların dikkatli ve verimli bir şekilde seçilmesi gerekmektedir. Bunu elde etmek için her bir stratejiye, karmaşıklık maliyetine bağlı olarak değişik bir ağırlık atanmaktadır. Bu ağırlık seçimi AIC ve MDL'nin [6], [7] karmaşıklık cezası ile aynı doğrultudadır.

II. Kaba Kuvvet Yaklaşımı ve Pişmanlık Sınırları

Bu bölümde, T uzunluğundaki bir oyun için M^T tane olası stratejilerin hepsinin paralel olarak çevrimiçi bir şekilde çalıştırıldığı varsayılmaktadır. Ancak, t anında, M^t tane paralel çalışan strateji bulunmaktadır. Bunlardan her biri farklı bir haydut kolunun kullanılmasını önermektedir. Bu stratejilerden her bir $\mathbf{s}_t$ için o stratejiye güveni gösteren $w_{\mathbf{s}_t}$ ağırlığı atanmaktadır. Bu ağırlıklara bağlı olarak, bir ihtimal simpleksi oluşturulmakta ve paralel çalışan her bir stratejiye ağırlıklar normalleştirilerek aşağıdaki gibi bir ihtimal değeri atanmaktadır:

$$P_{\mathbf{s}_t} = \frac{w_{\mathbf{s}_t}}{\sum_{\mathbf{s}'_t \in \mathbb{M}^t} w_{\mathbf{s}'_t}}, \tag{4}$$

denkleminde, $\mathbb{M}^t$ t anına kadarki stratejilerin sınıfıdır, ve büyüklüğü $|\mathbb{M}^t| = M^t$ şeklindedir. Her bir kol m için t anında seçim yapabilmek için t anında m'i öneren stratejiler bulunup onların olasılıkları aşağıdaki gibi toplanmaktadır:

$$p_{t,m} = \sum_{\mathbf{s}_t(t:t)=m} P_{\mathbf{s}_t}, \tag{5}$$

denkleminde, $\mathbf{s}_t(i:j)$ vektörü, $\mathbf{s}_t$'nin i'den j'ye kadar olan elemanlarını temsil etmektedir. Aynı haydut kolunu öneren stratejilerin ihtimallerini toplayarak her bir haydut kolunun t anındaki ihtimali oluşturulmaktadır. Denklem (4) ve (5)'deki hesaplamalar direkt olarak $w_{\mathbf{s}_t}$ ağırlıklarına bağlıdır. Ağırlık atanması iki bileşene sahiptir. İlk bileşen olarak, her bir $\mathbf{s}_t$'ye, sadece $\mathbf{s}_t$'nin karmaşıklığına bağlı öncül bir ağırlık atanmaktadır. İkinci kısım direkt olarak $\mathbf{s}_t$'nin geçmiş performansı olan $\exp(-\eta \tilde{L}_{\mathbf{s}_t(1:t-1)})$ üstel ağırlığına bağlıdır. Böylece, birleştirilmiş ağırlıklar şu şekildedir:

$$w_{\mathbf{s}_t} = \mathcal{T}(\mathbf{s}_t) e^{-\eta \tilde{L}_{\mathbf{s}_t(1:t-1)}}, \tag{6}$$

denkeleminde, η öğrenme hızıdır ve $\tilde{L}_{\mathbf{s}_t(1:t-1)}$, $L_{\mathbf{s}_t(1:t-1)}$ değişkeninin tarafsız tahmincisidir. Burada t anında sadece seçilen kolun kaybı olan l_{t,u_t} gözlemlenmektedir. Bu yüzden, diğer haydut kollarının kaybı için $\tilde{l}_{t,m}$ tahmini olulturulmaktadır. Bu amaç için iyi bilinen tarafsız tahminci aşağıdaki gibidir:

$$\tilde{l}_{t,m} = \begin{cases} l_{t,m}/p_{t,m} & m = u_t \\ 0 & m \neq u_t \end{cases} \tag{7}$$

denklemine göre, tahminin beklenen değeri gerçek değer eşittir $\mathbf{E}[\tilde{l}_{t,m}] = l_{t,m}$ [4]. Ayrıca burada, haydut kolları $m \in \{1, \ldots, M\}$ üzerinde $\mathbf{E}_m$ tanımlanmaktadır öyleki $\mathbf{E}_m[f(m)] = \sum_{m=1}^{M} p_{t,m} f(m)$ sağlanmaktadır. Böylece, $\mathbf{E}_m[\tilde{l}_{t,m}] = \sum_{m=1}^{M} p_{t,m} \tilde{l}_{t,m} = l_{t,u_t}$ eşitliği elde edilmektedir. Birleşim ağırlıkları $\mathcal{T}(\mathbf{s}_t)$ önceden belirlenmektedir. Tam anlamıyla çevrimiçi bir algoritma elde edebilmek için ardışık olarak hesaplanan $\mathcal{T}(\mathbf{s}_t)$ değerleri öyle bir seçilmektedirki iç içe geçme kuralı $\mathcal{T}(\mathbf{s}_t) = \mathcal{T}(\mathbf{s}_t|\mathbf{s}_t(1:t-1))\mathcal{T}(\mathbf{s}_t(1:t-1))$ elde edilmektedir. Burada, $\mathbf{s}_t(1:t-1)$ stratejisinden $\mathbf{s}_t$ stratejisine ağırlık güncellemesi $\mathcal{T}(\mathbf{s}_t|\mathbf{s}_t(1:t-1))$ ile gösterilmiştir. İhtimal skoru elde edebilmek için göreceli ağırlık güncellemeleri aşağıdaki eşitliği sağlayacak şekilde tasarlanmaktadır:

$$\sum_{m=1}^{M} \mathcal{T}([\mathbf{s}_t; m]|\mathbf{s}_t) = 1, \ \forall \mathbf{s}_t, \ \ t \in \{0, \ldots, T-1\}. \tag{8}$$

Yukarıda, $[\mathbf{s}_t; m]$ vektörü $\mathbf{s}_t$ vektörü ve m ($t+1$ uzunluğunda yeni bir strateji oluşturma) birleştirilmiş halidir, ayrıca, $\mathbf{s}_0 = [\emptyset]^T$ ve $\mathcal{T}(\mathbf{s}_0) = 1$ geçerlidir. Üstel ağırlıklar her bir stratejinin $t-1$ anına kadarki üstel kayıplarıdır. Bu yüzden, her bir stratejiye atanmış ortak ağırlık $w_{\mathbf{s}_t}$ ardışık olarak oluşturabilirdir öyleki

$$w_{\mathbf{s}_t} = w_{\mathbf{s}_t(1:t-1)} \mathcal{T}(\mathbf{s}_t|\mathbf{s}_t(1:t-1)) e^{-\eta \tilde{l}_{t,\mathbf{s}_t(t-1:t-1)}} \tag{9}$$

sağlanmaktadır.

Denklem (4), (5) ve (6)'daki haydut kol seçim olasılıkları kullanılarak, aşağıdaki pişmanlık sonucu elde edilmektedir.

Teorem 1: *$m \in \{1, ..., M\}$'in muhalif çok kollu haydutun kolları olduğunu ve $l_{t,m} \in [0,1]$ kaybının t anında m kolunu seçmekten dolayı oluşan kayıp olduğunu varsayalım. Denklem (8)'yı sağlayan ardışık olarak oluşturabilen herhangi bir birleşim ağırlık ataması $\mathcal{T}(\cdot)$ ve her bir kolun seçin ihtimalleri belirmek için olan (6)'daki gibi üstel kayıplar kullanılarak, aşağıdaki beklenen pişmanlık elde edilmektedir:*

$$\mathbf{E}[R_T] \leq \min_{\boldsymbol{s}_T} \left(\frac{\eta M T}{2} + \frac{1}{\eta} \ln W(\boldsymbol{s}_T) + L_{\boldsymbol{s}_T} - L_{\boldsymbol{s}^*_T} \right) \tag{10}$$

T raunt oyunda el edilmiştir. Burada, $\eta \geq 0$ üstel ağırlıklardaki öğrenme hızını ve $W(\boldsymbol{s}_T) \triangleq 1/\mathcal{T}(\boldsymbol{s}_T)$, $\boldsymbol{s}_T$ stratejisinin birleşim ağırlıklarının tersidir. $\boldsymbol{s}_T$ stratejisinin birkimli kaybı $L_{\boldsymbol{s}_T}$ ve $\boldsymbol{s}^*_T$ en iyi kol seçim stratejisinin birkimli kaybı $L_{\boldsymbol{s}^*_T}$ ile gösterilmektedir. En iyi strateji bütün kolların, $m \in \{1, ..., M\}$, bütün zaman indekslerindeki, $t \in \{1, ..., T\}$, kayıpları bilindiği öncül bilgisi ile seçilmiştir.

Teorem 1'deki sonuç dikkatli bir şekilde tasarlanmış $\mathcal{T}(\cdot)$ ve η ile altdoğrusal ve hatta en uygun pişmanlığın elde edilebileceğini göstermektedir. Ancak, ağırlık ataması $\mathcal{T}(\cdot)$'nun ardışık olarak oluşturabilir olması ve (8)'i sağlaması gerekmektedir. Buna ek olarak, Teorem 1'deki sonuç önerilen yapının performansının en iyi stratejinin karmaşıklık maliyetinin ($W(\mathbf{s}^*_T)$) yanısıra kaybı en iyi stratejinin kaybına (en iyi kayıp) göreceli olarak yakın olan stratejilerin karmaşıklık maliyetine de bağlı olduğunu göstermektedir. Bu yüzden, en iyi strateji yüksek karmaşıklık maliyetine sahip olsa bile, eğer en iyi kayba yeterince yakın düşük karmaşıklık maliyeti olan bir strateji varsa, önerilen algoritma göreceli olarak düşük bir pişmanlık elde edebilir. Denklem (10)'daki beklenti, sonuçların istatistiksel varsayımı olmayan haydut kayıplarının herhangi bir dizisi için eşit oranda geçerli olması için sahip olunan rastgelelik yüzündendir.

Teorem 1'in ispatı: t anında, en iyi seçim stratejisi $\mathbf{s}^*_T$'ye karşı olan pişmanlık $r_t = l_{t,u_t} - l_{t,s^*_t}$ ile gösterilmektedir. r_t daha idare edilebilir bir forma sokulup, iki farklı terim elde edilmektedir. Bu terimler ayrı ayrı olarak aşağıdaki gibi sınırlandırılmaktadır:

$$r_t = \left(l_{t,u_t} + \frac{\ln \mathbf{E}_m[e^{-\eta \tilde{l}_{t,m}}]}{\eta} \right) - \left(\frac{\ln \mathbf{E}_m[e^{-\eta \tilde{l}_{t,m}}]}{\eta} + l_{t,s^*_t} \right). \tag{11}$$

Denklem (11)'deki ilk terim $x > 0$ için $\ln x \leq x - 1$ kullanılarak aşağıdaki gibi sınırlandırılmaktadır:

$$\ln \mathbf{E}_m[e^{-\eta \tilde{l}_{t,m}}] \leq \mathbf{E}_m \left[e^{-\eta \tilde{l}_{t,m}} - 1 \right]. \tag{12}$$

Denklem (12), $x > 0$ için $e^{-x} - 1 + x \leq x^2/2$ kullanılarak aşağıdaki ifade elde edilmektedir:

$$\ln \mathbf{E}_m[e^{-\eta \tilde{l}_{t,m}}] \leq \mathbf{E}_m \left[\frac{\eta^2 \tilde{l}^2_{t,m}}{2} \right] - \eta \mathbf{E}_m[\tilde{l}_{t,m}] \leq \frac{\eta^2 l^2_{t,u_t}}{2 p_{t,u_t}} - \eta l_{t,u_t}. \tag{13}$$

Denklem (13) denklem (11)'deki ilk terimde yerine konulursa

$$r_t \le \frac{\eta}{2p_{t,u_t}} + \left[-\frac{1}{\eta}\ln \mathbf{E}_m[e^{-\eta \tilde{l}_{t,m}}] - l_{t,s_t^*}\right], \qquad (14)$$

$l_{t,u_t} \le 1$ olduğu için üstteki denklem elde edilmektedir. Denklem (14)'teki ikinci terimi üstten sınırlamak için (5) ve (4) kullanılarak aşağıdaki gibi beklenti hesaplanmaktadır:

$$\begin{aligned}\mathbf{E}_m[e^{-\eta\tilde{l}_{t,m}}] &= \sum_{m=1}^{M} p_{t,m} e^{-\eta \tilde{l}_{t,m}} = \sum_{\mathbf{s}'_t \in \mathbb{M}^t} P_{\mathbf{s}'_t} e^{-\eta \tilde{l}_{t,\mathbf{s}'_t(t:t)}}, \\ &= \sum_{\mathbf{s}'_t \in \mathbb{M}^t} \frac{w_{\mathbf{s}'_t}}{\sum_{\mathbf{s}''_t \in \mathbb{M}^t} w_{\mathbf{s}''_t}} e^{-\eta \tilde{l}_{t,\mathbf{s}'_t(t:t)}}, \\ &= \frac{\sum\limits_{\mathbf{s}'_t \in \mathbb{M}^t} \mathcal{T}(\mathbf{s}'_t) e^{-\eta \tilde{L}_{\mathbf{s}'_t}}}{\sum\limits_{\mathbf{s}''_t \in \mathbb{M}^t} \mathcal{T}(\mathbf{s}''_t) e^{-\eta \tilde{L}_{\mathbf{s}''_t(1:t\text{-}1)}}} = \frac{\sum\limits_{\mathbf{s}'_t \in \mathbb{M}^t} \mathcal{T}(\mathbf{s}'_t) e^{-\eta \tilde{L}_{\mathbf{s}'_t}}}{\sum\limits_{\mathbf{s}''_{t\text{-}1} \in \mathbb{M}^{t\text{-}1}} \mathcal{T}(\mathbf{s}''_{t\text{-}1}) e^{-\eta \tilde{L}_{\mathbf{s}''_{t\text{-}1}}}},\end{aligned} \qquad (15)$$

burada, (6) ve (8) kullanılmaktadır. Bundan sonra, (15)'in logaritmalarını bütün T rauntları için toplayıp, $-1/\eta$ ile çarparak aşağıdaki denklem elde edilmektedir:

$$\begin{aligned}\sum_{t=1}^{T} -\frac{1}{\eta} \ln \mathbf{E}_m[e^{-\eta \tilde{l}_{t,m}}] &= -\frac{1}{\eta} \ln \sum_{\mathbf{s}'_T \in \mathbb{M}^T} \mathcal{T}(\mathbf{s}'_T) e^{-\eta \tilde{L}_{\mathbf{s}'_T}}, \\ &\le -\frac{1}{\eta} \ln \left[\mathcal{T}(\mathbf{s}_T) e^{-\eta \tilde{L}_{\mathbf{s}_T}}\right] \le -\frac{1}{\eta} \ln \mathcal{T}(\mathbf{s}_T) + \tilde{L}_{\mathbf{s}_T}.\end{aligned} \qquad (16)$$

Yukarıdaki denklem herhangi bir $\mathbf{s}_T \in \mathbb{M}^T$ için geçerlidir. Denklem (14) bütün T rauntları için toplanarak aşağıdaki gibi T rauntluk oyunda birikmiş pişmanlık değeri bulunmaktadır:

$$R_T = \sum_{t=1}^{T} \frac{\eta}{2p_{t,u_t}} - \frac{1}{\eta} \sum_{t=1}^{T} \ln \mathbf{E}[e^{-\eta \tilde{l}_{t,m}}] - \sum_{t=1}^{T} l_{t,s_t^*}. \qquad (17)$$

Denklem (16)'yı denklem (17)'de kullanarak, toplam pişmanlık

$$R_T \le \sum_{t=1}^{T} \frac{\eta}{2p_{t,u_t}} - \frac{1}{\eta} \ln \mathcal{T}(\mathbf{s}_T) + \tilde{L}_{\mathbf{s}_T} - L_{\mathbf{s}_T^*} \qquad (18)$$

şeklinde yazılır. Seçim u_t, ve böylece p_{t,u_t}, R_T'deki rastgele değişkenlerdir. Denklem (18)'in kol seçim ihtimallerine göre beklentisi alındığında

$$\mathbf{E}[R_T] \le \frac{\eta M T}{2} - \frac{1}{\eta} \ln \mathcal{T}(\mathbf{s}_T) + L_{\mathbf{s}_T} - L_{\mathbf{s}_T^*}$$

elde edilmektedir. Gösterim kolaylığı için birleşim ağırlıklarından stratejinin karmaşıklık maliyetine kadar olan gösterimler $W(\mathbf{s}_T) \triangleq 1/\mathcal{T}(\mathbf{s}_T)$ olacak şekilde değiştirilmektedir. Böylece,

$$\mathbf{E}[R_T] \le \frac{\eta M T}{2} + \frac{1}{\eta} \ln W(\mathbf{s}_T) + L_{\mathbf{s}_T} - L_{\mathbf{s}_T^*} \qquad (19)$$

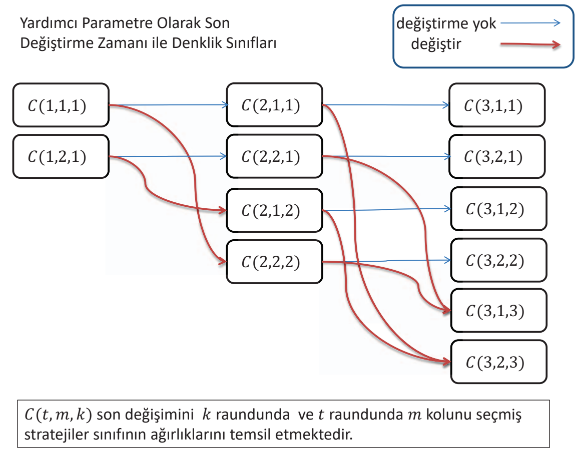
elde edilmektedir. Denklem (19) herhangi bir strateji $\mathbf{s}_T$ için sağlandığı için daha sıkı bir sınır (19)'u $\mathbf{s}_T$ üzerinden en küçük duruma getirerek elde edilebilir. Ve bu işlem (10)'u verir. $\square$

Sonuç 1: *Kayıp dizisinden bağımsız bir üst sınır elde etmek için Teorem 1'de $\mathbf{s}_T = \mathbf{s}_T^*$ eşitliği kullanılarak* (10) *denklemi aşağıdaki gibi üstten sınırlandırılmaktadır:*

$$\mathbf{E}[R_T] \le \frac{\eta M T}{2} + \frac{1}{\eta} \ln W(\mathbf{s}_T^*). \qquad (20)$$

Yukarıdaki denklemde, $W(\mathbf{s}_T^*)$ en iyi kol seçim stratejisi $\mathbf{s}_T^*$'nin birleşim ağırlıklarının tersidir.

III. VERİMLİ UYGULAMA



Şekil 1: Son değiştirme zamanını yardımcı parametre olarak kullanarak, iki kollu haydut durumunun ilk 3 raundu için verimli birleşim örneği. Bu durumda, yardımcı parametre vektörü $\boldsymbol{\sigma}_t$ sadece k'yi içermektedir ve alabileceği olası değerler zaman ile doğrusal olarak artmaktadır. Bu yüzden, k sadece t değişik değer alabildiği için bu yapı doğrusal karmaşıklığa sahip bir algoritmayı formülleştirmektedir.

Hesaplama karmaşıklığını azaltmak için belli stratejileri beraber gruplayarak denklik sınıfları oluşturulmaktadır. $C(t, m, \boldsymbol{\sigma}_t)$ m kolunun denklik sınıfının ve t anındaki $\boldsymbol{\sigma}_t$ yardımcı parametresinin ağırlığı olarak tanımlanmaktadır. Denklik sınıfı $C(t, m, \boldsymbol{\sigma}_t)$ t anında m kolunu seçen bütün $\mathbf{s}_t$ stratejilerini içermektedir ve davranışı $\boldsymbol{\sigma}_t$ parametre vektörü ile eşleşmektedir. Örnek olarak, Şekil 1'deki $\boldsymbol{\sigma}_t$ vektörünün sadece stratejilerin yaptığı son değiştirmenin zaman indeksini içermesi düşünülebilir. Stratejileri son değiştirme zamanına göre gruplandırmak, doğrusal olarak artan sayıda denklik sınıflarına neden olmaktadır. Yardımcı parametre $\boldsymbol{\sigma}_t$ farklı şekilde gruplar da içerebilir. Örnek olarak, stratejilerin yaptığı değişim sayısı verilebilir.

$\boldsymbol{\sigma}_t$'ya dahil edilen parametreler, onun en sonda kaç tane strateji belirleyeceğini ve kaç tane denklik sınıfına sahip olacağını belirlemektedir. Yardımcı parametre $\boldsymbol{\sigma}_t$ kullanılmasının sebebi (9)'daki ağırlık güncellemeleri aynı olan belli stratejileri gruplamaktır. Bu yüzden, $\boldsymbol{\sigma}_t$'ya birleşim ağırlık güncellemeleri, $\mathcal{T}(\mathbf{s}_t|\mathbf{s}_t(1\!:\!t\!-\!1))$, ile alakalı bütün parametreler dahil edilmelidir. Böylece, bileşim ağırlık atanmasının, $\mathcal{T}(\cdot)$, tasarlanması $\boldsymbol{\sigma}_t$'ya dahil edilecek parametreleri etkilemektedir. Burada, Λ_t olası bütün $\boldsymbol{\sigma}_t$ vektörlerini içeren vektör uzayı olarak tanımlanmaktadır.

Denklik sınıfının ağırlığı, sınıf parametreleri $t, m, \boldsymbol{\sigma}_t$ ile uyumlu davranan stratejilerin ağırlıkları toplamıdır. Buna göre aşağıdaki denklem elde edilmektedir.

$$C(t, m, \boldsymbol{\sigma}_t) = \sum_{\substack{\mathbf{s}_t(t:t)=m \\ \boldsymbol{\sigma}(\mathbf{s}_t)=\boldsymbol{\sigma}_t}} w_{\mathbf{s}_t}. \qquad (21)$$

Yukarıda, $\boldsymbol{\sigma}(\cdot)$ $\mathbf{s}_t$'den $\boldsymbol{\sigma}_t$ parametresine kadar olan eşleme fonksiyonudur ve $\boldsymbol{\sigma} : \mathbb{M}^t \rightarrow \Lambda_t$ olarak tanımlanmaktadır. Ayrıca, $w_{\mathbf{s}_t}$ (6)'da tanımlanmaktadır. Şekil 1'de, iki kollu haydut oyununun ilk üç raundu için denklik sınıfı örneği verilmiştir. Bu şekilde, $C(t, m, k)$ son değiştirmeyi k raundunda yapmış ve t raundunda m kolunu seçmiş stratejilerin ağırlıklarını temsil etmektedir. Örnek olarak, $C(3, 1, 3)$ $\mathbf{s}_3 \in \{[2, 2, 1]^T, [1, 2, 1]^T\}$ stratejiler sınıfının ağırlığıdır. Buradaki amaç (9)'daki çarpımsal güncellemeyi belli sayıda stratejiler için aynı anda yapmak olduğu için, $\mathcal{T}(\mathbf{s}_t|\mathbf{s}_t(1\!:\!t{-}1))\exp(-\eta\tilde{l}_{t,\mathbf{s}_t(t-1:t-1)})$ güncellemesinin karışımdaki bütün stratejiler için aynı olması gerekmektedir. Üstel kayıp güncellemesi eğer stratejiler tarafından seçilen kollar aynı ise aynıdır. Bu koşul ancak stratejiler aynı denklik sınıfına ait olduğunda gerçekleşmektedir. Birleşim ağırlıkları güncellemeleri sınıf parametreleri olan şimdiki raunt t, şimdiki raunttaki kol seçimi m ve yardımcı parametre $\boldsymbol{\sigma}_t$'ya bağlı olarak tasarlanmaktadır öyleki aynı sınıfta olan stratejiler aynı ağırlık güncellemesine sahiptirler. Denklik sınıfı $C(t,m',\boldsymbol{\sigma}'_t)$'den $C(t+1,m,\boldsymbol{\sigma}_{t+1})$'ye kadar olan ortak birleşim ağırlık güncellemesi $\mathcal{T}(t+1,m,\boldsymbol{\sigma}_{t+1}|t,m',\boldsymbol{\sigma}'_t)$ ile gösterilmektedir. Bu gösterimde, m' ve $\boldsymbol{\sigma}'_t$ sonraki zaman indeksleri arasında ayrım yapmak için kullanılmaktadır. Bu yüzden,

$$C(t{+}1,m,\boldsymbol{\sigma}_{t+1}) = \sum_{m',\boldsymbol{\sigma}'_t} C(t,m',\boldsymbol{\sigma}'_t)\mathcal{T}(t{+}1,m,\boldsymbol{\sigma}_{t+1}|t,m',\boldsymbol{\sigma}'_t)e^{-\eta\tilde{l}_{t,m'}}, \quad (22)$$

her bir denklik sınıfı ağırlığı kendi parametrelerine uyan stratejilerin ortak ağırlığının toplamı olarak hesaplandığı için yukarıdaki denklem geçerli olmaktadır.

Üstel kayıp güncellemesi son seçilen kola bağlı olduğu için, denklik sınıfları stratejileri son seçtikleri kola göre gruplamaktadır. $\boldsymbol{\sigma}_t$'daki yardımcı parametre birleşim ağırlıklarını güncellemek için kullanılmaktadır. Algoritma 1'de, genel yapının tam verimli uygulaması sunulmaktadır.

Denklem (22) , (21)'i kullanılarak (9) denkleminin direkt bir uygulaması olduğu için, Algoritma 1'deki verimli uygulama, direkt olarak Bölüm II'deki kaba kuvvet yaklaşımının ağırlık atamasını uygulamaktadır. Bu yüzden, kaba kuvvet yaklaşımı için yapılan bütün pişmanlık analizleri (Teorem 1 ve Sonuç 1 gibi) Algoritma 1'deki verimli uygulama için de geçerlidir. Hesaplama karmaşıklığı denklik sınıfı sayısı (t anında $M|\Lambda_t|$ ile gösterilmektedir.) ile alakalı olduğu için, denklik sınıfları kullanılarak hesaplama karmaşıklığı zaman içinde üstel olmaktan terimsel olmaya indirgenebilmektedir.

Açıklama 1: Yardımcı parametre $\boldsymbol{\sigma}_t$ kullanılarak genel ağırlık atamasında daha fazla esneklik sağlanmaktadır. Bu yapının genelliği ve ağırlık ataması çeşitli uygulamalar için bir çok ihtimal sağlamaktadır. Değişik ortamlar için değişik ağırlık atamaları tasarlanabilir. Ağırlık atamaları değişik karmaşıklıktaki maliyet fonksiyonlarına uygun hala getirilebilir. Örnek olarak, bütün değişimlere eşit davranmak yerine uzun bölümlerden sonraki değişimlere daha fazla önem verilmesi düşünülebilir. Eğer bölümler belli uzunluktan kısa ise aykırı olarak düşünülüp, değişim olarak görülmeyebilir. Bu örnek için son değişim zamanı yardımcı değişken olarak kullanılması ile uygun bir ağırlıklandırma şeması tasarlanabilir. Buna ek olarak, bu genel yapı, bütün küme $\mathbb{M}^T$ yerine sadece stratejilerin makul bir alt kümesini birleştirmek için kullanılabilir. Örnek olarak, eğer en iyi kol en azından K raunt için değişmiyorsa, son parçanın uzunluğu yardımcı değişken olarak kullanılabilir ve parçalar K uzunluğuna ulaşmadan olan değişimleri engelleyerek, sadece en azından K uzunluğundaki parçaya sahip stratejiler birleştirilebilir. Eğer belli bir kol m en iyi kol olan m''dan hemen sonraki en iyi kol olmazsa, sadece makul stratejileri birleştirmek için m''dan m'e olan değişimleri engelleyen bir ağırlıklandırma şeması tasarlanabilir.

Algorithm 1 Verimli Genel Yapı

Sabit $\eta \in \mathbb{R}^+$'ya ilk değer ata
Birleşim ağırlık atamalarını seç
$t \in 1, ..., T$ için Λ_t'yi belirle
Λ_1'in uzantısı olan $\boldsymbol{\sigma}_1$'ya ilk değer atama
$m \in 1, ..., M$ için $C(1, m, \boldsymbol{\sigma}_1) = 1/M$'yi belirle
$p_{1,m} = C(1, m, \boldsymbol{\sigma}_1)$ için ilk değer ata
for $t = 1 : T$ **do**
    $p_{t,m}$ ihtimali ile M koldan birini seç
    Kayıp l_{t,u_t}'nin alınması
    $m \in 1, ..., M$ için $\tilde{l}_{t,m} = \frac{l_{t,m}\mathbb{1}_{m=u_t}}{p_{t,m}}$'yi belirle
    for $\boldsymbol{\sigma}_{t+1} \in \Lambda_{t+1}$ **do**
        for $m = 1 : M$ **do**
            $C(t{+}1,m,\boldsymbol{\sigma}_{t+1}) = \sum_{m',\boldsymbol{\sigma}'_t} C(t,m',\boldsymbol{\sigma}'_t)\mathcal{T}(t{+}1,m,\boldsymbol{\sigma}_{t+1}|t,m',\boldsymbol{\sigma}'_t)e^{-\eta\tilde{l}_{t,m'}}$
        end for
    end for
    for $m = 1 : M$ **do**
        $p_{t+1,m} = \frac{\sum_{\boldsymbol{\sigma}_{t+1}\in\Lambda_{t+1}} C(t+1,m,\boldsymbol{\sigma}_{t+1})}{\sum_{m=1}^{M}\sum_{\boldsymbol{\sigma}_{t+1}\in\Lambda_{t+1}} C(t+1,m,\boldsymbol{\sigma}_{t+1})}$'yi belirle
    end for
end for

IV. SONUÇLAR

Bu bildiride, muhalif çok kollu haydut problemi çalışılmış ve genel olarak uygulanabilen verimli bir haydut kol seçim yapısı önerilmiştir. Çeşitli uygulamalar için, önerilen yapı her türlü ağırlıklandırma şeması ile çalışmaktadır. Bu yapı ardışık olarak bütün olası kol seçim stratejilerini dikkatli bir şekilde oluşturulmuş ağırlıklar ile birleştirmektedir. Burada, olası strateji sayısı M^T ile büyümektedir. Belirli stratejileri gruplandırıp topluca güncelleyen denklik sınıfları yaratarak, bu ağ yapısı verimli bir şekilde uygulanmıştır.

KAYNAKLAR

[1] V. Krishnamurthy and R. J. Evans, "Hidden markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking," *Signal Processing, IEEE Transactions on*, vol. 49, no. 12, pp. 2893–2908, 2001.

[2] ——, "Correction to" hidden markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking"." *IEEE Transactions on Signal Processing*, vol. 51, no. 6, pp. 1662–1663, 2003.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, Oct 1995, pp. 322–331.

[4] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, Jan. 2003.

[6] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, Dec 1974.

[7] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.