Modele Dayalı Pekiştirme ile Öğrenme için Ardışık Monte Carlo Örnekleyicileri Sequential Monte Carlo Samplers for Model-Based Reinforcement Learning

Orhan Sönmez, A. Taylan Cemgil
Bilgisayar Mühendisliği Bölümü
Boğaziçi Üniversitesi
İstanbul, Türkiye
Email: orhan.sonmez,taylan.cemgil@boun.edu.tr

***Özetçe* —Pekiştirme ile öğrenme problemi, genel olarak Bellman denklemleri çözümlerinin kısmi eniyi sonuçlarına sabit noktalı yineleme metodları ile yaklaşarak çözülmektedir. Ancak, bu problemi eşdeğer bir olabilirlik enbüyütme problemine çevirmek ve olasılıksal çıkarım yöntemlerini bu problemin çözümünde kullanmak da mümkündür. Biz de modele dayalı biçim çözümü için beklenti adımında Metropolis-Hastings çekirdekli ardışık Monte Carlo örnekleyicileri kullanan bir beklenti-enbüyütme algoritması önerdik. Sonra da algoritmamızı ölçüt pekiştirme ile öğrenme problemlerinden dağ-araba problemi üzerinde değerlendirdik.**

Anahtar Kelimeler*—*Ardışık Monte Carlo Örnekleyicileri, Pekiştirme ile Öğrenme, Markov Karar Süreçleri, Beklenti-Enbüyütme, Metropolis-Hastings.

***Abstract*—Reinforcement learning problems are generally solved by using fixed-point iterations that converge to the suboptimal solutions of Bellman equations. However, it is also possible to formalize this problem as an equivalent likelihood maximization problem and employ probabilistic inference methods. We proposed an expectation-maximization algorithm that utilizes sequential Monte Carlo samplers with Metropolis-Hastings kernels in its expectation step to solve the model-based version. Then, we evaluate our algorithm on mountain-car problem which is a benchmark reinforcement learning problem.**

Keywords*—*Sequential Monte Carlo Samplers, Reinforcement Learning, Markov Decision Processes, Expectation-Maximization, Metropolis-Hastings.

I. GİRİŞ

Pekiştirme ile öğrenme problemi, Markov karar süreçleri üzerinde tanımlanan genel bir kontrol problemdir. Fakat, yapısı gereği çok özel durumlar haricinde kapalı biçim bir çözümü bulunmamaktadır [1]. Bu yüzden, pekiştirme ile öğrenme problemi genelde yaklaşık olarak Bellman denklemleri çözümlerine yakınsayarak çözülmektedir [2],[3]. Ancak, bu bakış açısının yanı sıra bu problemi eşdeğer bir olabilirlik enbüyütme problemine çevirmek ve olasılıksal çıkarım yöntemlerini bu problemin çözümünde kullanmak da mümkündür.

978-1-4673-5563-6/13/$31.00

İlk olarak Toussaint ve Storkey [4] Markov karar süreçleri üzerinde bir karışım modeli tanımlayarak pekiştirme ile öğrenme problemine eşdeğer bir olabilirlik enbüyütme problemi sunmuştur. Aynı zamanda bu problemin çözümü için de tam tamına çıkarım yapan bir beklenti-enbüyütme algoritması türetmiştir. Daha sonra, Furmston ve Barber [5] karışım modelinin Markov özelliklerinden faydalanıp bu tam tamına çıkarım metodunu iyileştirmişlerdir.

Ancak, problemin durum-eylem uzayı büyüdükçe tam tamına çıkarım yapmak üssel olarak zorlaşmakta ve pratik olarak kullanılamaz hale gelmektedir. Bu yüzden, Sönmez ve Cemgil [6] önem örneklemesi ve Hoffman vd. [7] ise tersinir atlama Markov zinciri Monte Carlo kullanarak yaklaşık olarak çıkarım yapmışlardır.

Biz ise, pekiştime ile öğrenme probleminin çözümü için türetilmiş bu beklenti-enbüyütme algoritmasının [4] beklenti adımında kullanılmak üzere Metropolis-Hastings çekirdekli ardışık Monte Carlo örnekleyicileri öneriyoruz.

Bildirinin devamında, II. bölümde Markov karar süreçleri üzerinde pekiştirme ile öğrenme problemini tanımlıyoruz ve sonra III. bölümde de bu problemin çözümü için önerdiğimiz yöntemi sunuyoruz. Son olarak ise, IV. bölümde yöntemimizi ölçüt bir problem üzerinde gerçeklediğimiz deneyi ve sonuçlarını ve de V. bölümde de vargılarımızı ve gelecek çalışmalarımızı sunuyoruz.

II. PROBLEM

A. Markov Karar Süreçleri

Markov karar süreçleri (MKS), bir ortamda fayda tabanlı karar veren ajanların ardışık karar verme süreçlerini modellemek için kullanılan olasılıksal çerçevelerdir. Bu süreç boyunca, ajan her t anında bir $x_t \in \mathcal{X}$ durumunda bulunur. Daha sonra, ajan π ilkesi ve içinde bulunduğu x_t durumuna göre bir $a_t \in \mathcal{A}$ eylemini gerçekleştirir ve bunun sonucu olarak da bir r_t ödülü alır ve $t+1$ anı için bir x_{t+1} durumuna geçer.

Yani biçimsel olarak $t = 0, 1, ..., T$ zaman adımları için tanımlanmış bir MKS aşağıdaki olasılık modeline göre işler

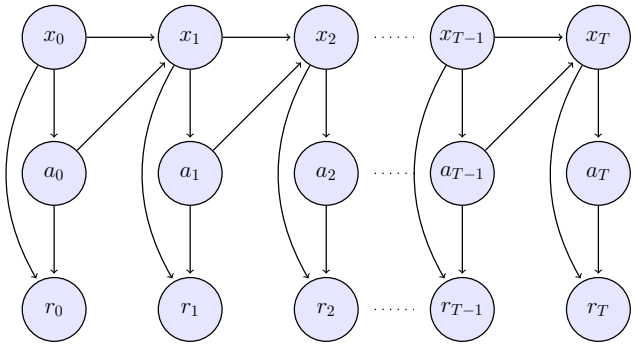
ve Şekil 1'deki grafik modeline sahiptir.

$$\begin{aligned} x_0 &\sim P(x_0) \\ a_t &\sim P(a_t|x_t;\pi) \\ r_t &\sim P(r_t|x_t,a_t) \\ x_{t+1} &\sim P(x_{t+1}|x_t,a_t) \end{aligned} \tag{1}$$

Buna göre de, belirli bir π ilkesi için T uzunluğundaki herhangi bir durum-eylem gezingesi $x_{0:T}, a_{0:T}$ verilen bir MKS'den,

$$\begin{aligned} P(x_{0:T}, a_{0:T}|T;\pi) =& P(x_0)P(a_T|x_T;\pi) \\ & \cdot \prod_{t=0}^{T-1} P(a_t|x_t;\pi)P(x_{t+1}|x_t,a_t) \end{aligned} \tag{2}$$

şeklindeki önsel dağılıma göre gelmektedir.



Şekil 1: Markov karar süreci grafik modeli

B. Pekiştirme ile Öğrenme

Pekiştirme ile öğrenme (PÖ) problemi ise, MKS ile modellenmiş bir ajanın toplam ödülünü enbüyüten ilkeyi bulmak olarak tanımlanır. MKSler olasılıksal süreçler olduğu için de, toplam ödülün tüm durum-eylem gezingeleri üzerinden beklenen değerinin hesaplanması gerekmektedir.

Ayrıca, bunun yanı sıra MKSnin tanımlı olduğu T zaman indisinin sonsuza gittiği koşullarda, toplam ödül değerinin ıraksamaması için bir $0 < \gamma < 1$ indirim faktörü tanımlamakta ve ödüller de bu indirim faktörüne göre üssel olarak azaltılmaktadır.

Böylece, genel bir PÖ problemi,

$$\pi^* = \arg\max_{\pi} \left\langle \sum_{t=0}^{T} \gamma^t r_t \right\rangle \tag{3}$$

beklenen değer denklem (2)'deki önsel dağılıma göre hesaplanmak suretiyle yukarıdaki biçimde tanımlanmaktadır. Problemin çözümü olan π^* da eniyi ilke olarak ifade edilir.

III. YÖNTEM

A. Beklenti-Enbüyütme

Toussaint ve Storkey [4], denklem (3)'deki PÖ problemini klasik yöntemlerle çözmek yerine, ona eşdeğer bir olabilirlik enbüyütme problemi önermiştir. Bunun için, her biri ayrı ayrı $t = 0, 1, ..., T$ uzunluğunda olan MKSler üzerinde,

$$P(T = t) \propto \gamma^t \tag{4}$$

önsel dağılımına göre bir karışım modeli tanımlamıştır.

Daha sonra da, bu problemi çözmeye yönelik bir ilke yineleme yöntemine karşılık gelen bir beklenti-enbüyütme (BE) algoritması türetmişlerdir. Bu algoritmaya göre, rastgele bir $\pi^{(0)}$ ilkesi ile başlandıktan sonra BE algoritmasının her k adımında $(k+1)$. adımdaki ilke $\pi^{(k+1)}$,

$$\pi^{(k+1)} \leftarrow \arg\max_{\pi} \langle \log P(r = 1, x_{0:T}, a_{0:T}, T; \pi) \rangle \tag{5}$$

beklenen değer,

$$P(x_{0:T}, a_{0:T}, T|r = 1; \pi^{(k)}) \tag{6}$$

sonsal dağılımına göre olmak üzere elde edilir. Bu yineleme işlemi de ilke yakınsayıncaya kadar tekrarlanır.

Bunun yanı sıra, MKSlerin Markov özelliğinden dolayı denklem (5) ile tanımlanmış olan enbüyütme problemi kapalı biçim bir çözüme sahiptir. Herhangi bir π ilkesi,

$$\pi_{i,a} \equiv P(a_t = a|x_t = i;\pi) \tag{7}$$

parametreleri ile ifade edilirse üzere, $(k+1)$. adımdaki ilkenin parametreleri $\pi_{i,a}^{(k+1)}$, beklenen değerler denklem (6)'ya göre hesaplanmak üzere aşağıdaki biçimde elde edilmektedir.

$$\pi_{i,a}^{(k+1)} = \frac{\left\langle \sum_{t=0}^{T} [x_t = i \wedge a_t = a] \right\rangle}{\left\langle \sum_{t=0}^{T} [x_t = i] \right\rangle} \tag{8}$$

B. Ardışık Monte Carlo Örnekleyicileri

Bölüm III-A'da sunulan BE algoritmasını gerçeklemek için denklem (8)'deki beklenen değerler hesaplanmalıdır. Ancak, problemin boyutu arttıkça bu değerleri tam tamına hesaplamak pratikte mümkün olmayacaktır. Ancak, beklenen değerlerin hesaplandığı denklem (6)'deki sonsal dağılımdan S adet örnek çekildiği varsayılırsa, beklenen değerleri denklem (9) biçiminde ağırlıklandırılmış bir Monte Carlo tahmini ile yakınsamak mümkün olabilmektedir [6].

$$\begin{aligned} \left\langle \sum_{t=0}^{T} [x_t = i \wedge a_t = a] \right\rangle &\approx \sum_{s=1}^{S} w(x_{0:T}^{(s)}, a_{0:T}^{(s)}) \sum_{t=0}^{T} [x_t^{(s)} = i \wedge a_t^{(s)} = a] \\ \left\langle \sum_{t=0}^{T} [x_t = i] \right\rangle &\approx \sum_{s=1}^{S} w(x_{0:T}^{(s)}, a_{0:T}^{(s)}) \sum_{t=0}^{T} [x_t^{(s)} = i] \end{aligned} \tag{9}$$

Sönmez ve Cemgil [6], önem örneklemesi ve Hoffman vd. [7] de tersinir atlama Markov zincir Monte Carlo yöntemlerini kullanarak bu beklenen değerleri yaklaşık olarak kestirmişlerdir. Biz de, en gelişkin olasılıksal yaklaşık çıkarım yöntemlerinden birisi olan ardışık Monte Carlo örnekleyicileri (AMCÖ) [8] kullanarak bu kestirimi yapan bir algoritma öneriyoruz.

Algoritmamızın detaylarına girmeden önce, her biri T uzunluğunda birer durum-eylem gezingesine denk gelen örneklerimizi, simgelemi basitleştirmek adına z olarak adlandırıyoruz.

$$z \equiv (x_{0:T}, a_{0:T}, T) \tag{10}$$

1) Köprü Fonksiyonları: İlk etapla, ardışık olarak örnekleme yapacağımız N adet köprü fonksiyonu tanımlıyoruz. Buna göre, $n = 1, 2, ..., N$ değerleri için, $\phi_n(z_n)$ köprü fonsiyonlarını,

$$\phi_n(z_n) \propto P(z_n; \pi) P(r = 1 | z_n; \pi)^{\eta(n)} \tag{11}$$

şeklinde tanımladıktan sonra köprü fonksiyonunu karakterize eden $\eta(\cdot)$ üs fonksiyonunu bir nevi tavlama mekanizması olarak kullanmak üzere,

$$0 \equiv \eta(1) < \eta(2) < ... < \eta(N) \equiv 1 \tag{12}$$

biçiminde monotonik ve artan bir fonksiyon olarak seçiyoruz.

Böylelikle, ilk olarak örnekleme yapacağımız köprü fonksiyonu $\phi_0(z_0)$ aslında denklem (1)'deki biçimde rahatlıkla örnekleme yapabileceğimiz durum-eylem gezingelerinin önsel dağılımına eşit oluyor. Gene benzer şekilde son köprü fonksiyonu $\phi_N(z_N)$ de hedef olarak örnekleme yapmaya çalıştığımız denklem (6)'daki durum-eylem gezingelerinin sonsal dağılımına denk geliyor.

Yani, yukarıda tanımladığımız köprü fonksiyonlarından ardışık olarak örnekleme yaparak, ϕ_N köprü fonksiyonundan çektiğimiz z_N örneklerini kullanarak denklem (9)'daki biçimde denklem (8)'deki beklenen değerleri hesaplıyoruz.

2) İleri-Geri Çekirdekler: AMCÖ ile örnekleme yapabilmek için ardışık köprü fonksiyonları arasında $n = 2, 3, ..., N$ olmak üzere $K_n(z_n|z_{n-1})$ ileri çekirdek tanımlamamız gerekiyor. Biz de, örneklerimizin verimliliğini olabildiğince yüksek tutmak için K_n ileri çekirdeklerini asimtotik olarak ϕ_n dağılımından örnekleme yapan Metropolis-Hastings (MH) çekirdeği olarak seçtik.

Benzer bi şekilde, $n = 2, 3, ..., N$ için $L_{n-1}(z_{n-1}|z_n)$ geri çekirdeklerini de tanımlamamız gerekiyor. Onları da daha sonra ağırlık fonksiyonunu kapalı biçimde hesaplayabilmek için, ileri çekirdeklere bağlı olarak tanımladık. Böylece, her L_{n-1} çekirdeğini K_n ile aynı olacak şekilde asimtotik olarak ϕ_n dağılımından örnekleme yapan MH çekirdeği olarak seçtik.

Son olarak da, her iki çekirdekte de kullanılmak üzere verilen bir ϕ_n için K_n ileri çekirdeğine denk gelen MH çekirdeğini aşağıdaki biçimde türettik. Buna göre $q(\tilde{z}_n|z_{n-1})$ teklif fonksiyonunu,

$$\begin{aligned} z_{n-1} &\equiv (x_{0:T}, a_{0:T}, T) \\ \tau &\sim U[1...T] \\ (\tilde{x}_{0:\tau}, \tilde{a}_{0:\tau-1}) &= (x_{0:\tau}, a_{0:\tau-1}) \\ \tilde{a}_t &\sim P(a_t|\tilde{x}_t; \pi) \qquad \text{for } t = \tau..T \\ \tilde{x}_{t+1} &\sim P(x_{t+1}|\tilde{x}_t, \tilde{a}_t) \qquad \text{for } t = \tau..T-1 \\ \tilde{z}_n &\equiv (\tilde{x}_{0:T}, \tilde{a}_{0:T}, T) \end{aligned} \tag{13}$$

kullandık. Bu teklif fonksiyonu için kabul olasılığını en sadeleşmiş biçimiyle,

$$\alpha(z_{n-1} \to \tilde{z}_n) = \min\left\{1, \frac{P(r = 1|\tilde{z}_n)^{\eta(n)}}{P(r = 1|z_{n-1})^{\eta(n)}}\right\}$$

şeklinde türettik.

3) Ağırlık Fonksiyonu: Köprü fonksiyonları ve ileri-geri çekirdek seçimlerize göre de, AMCÖ ile örnekleme sırasında herhangi bir n için $z_{0:n}^{(s)} \equiv (z_0^{(s)}, ..., z_n^{(s)})$ ardışık örneklemesinin özyinelemeli ağırlık fonksiyonunu da kapalı biçimde,

$$W(z_{0:n}^{(s)}) = W(z_{0:n-1}^{(s)}) \frac{\phi_n(z_n^{(s)})}{\phi_{n-1}(z_{n-1}^{(s)})} \tag{14}$$

olarak elde ettik. Daha sonra da, Monte Carlo tahmininde kullanılacak durum-eylem gezingeleri $z_N^{(s)}$ için normalize edilmiş marjinal ağırlıkları,

$$w(z_N^{(s)}) = \frac{W(z_{0:N}^{(s)})}{\sum_{s'=0}^{S} W(z_{0:N}^{(s')})} \tag{15}$$

şeklinde hesapladık. Ardışık örnekleme sonlandığı zaman da bu ağırlıklara göre denklem (9)'daki biçimde beklenen değerleri yaklaşık olarak kestirdik.

IV. DENEYLER VE SONUÇLAR

Önerdiğimiz yöntemi PÖ probleminin ölçüt problemlerinden dağ-araba problemi [9] üzerinde değerlendirdik. Doğrusal olmayan bir geçiş modeline sahip olduğu için problemin kapalı biçim bir çözümü bulunmamakta ve dolayısıyla da çıkarım metotlarına ihtiyaç duyulmaktadır.

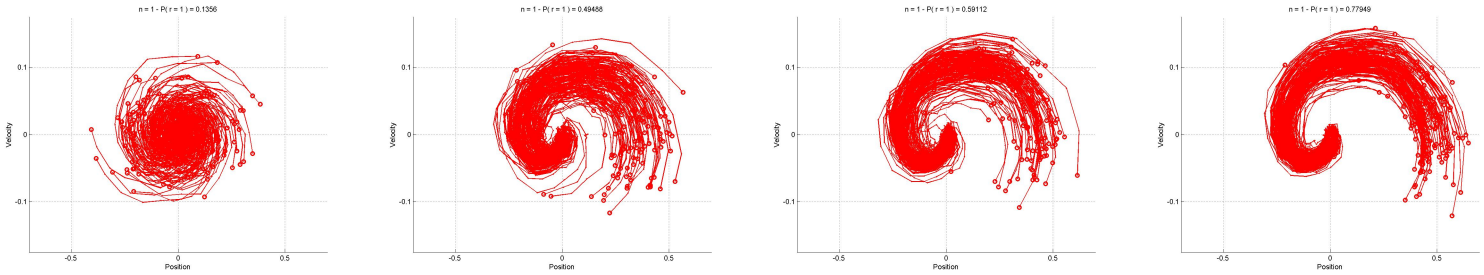
Aynı zamanda sürekli bir durum uzayına sahip olduğundan ilkeyi temsil edebilmek için ya durum uzayını ayrıklaştırmak ya da ilkeyi duruma bağlı bir fonksiyon olarak ele alıp, o fonksiyonu kestirmek gerekmektedir. Biz ise bu sorun için, bir çok yapay öğrenme probleminde yüksek başarımla çalışan k en yakın komşu (k-EYK) [10] yöntemini kullanarak dolaylı yoldan bir ayrıklaştırma sağladık.

Durum uzayı sürekli olduğundan denklem (8)'i sağlayan sonsuz sayıda k-EYK çözümü olabilmektedir. Ancak, BE adımı sonunda hedef dağılımdan örneklenen durum-eylem gezingeleri de bir çözümlerden birine denk gelmektedir. Biz de, doğrudan bu gezingelerle bir sonraki BE yinelemesinde kullanılacak olan ilkeyi niteledik.

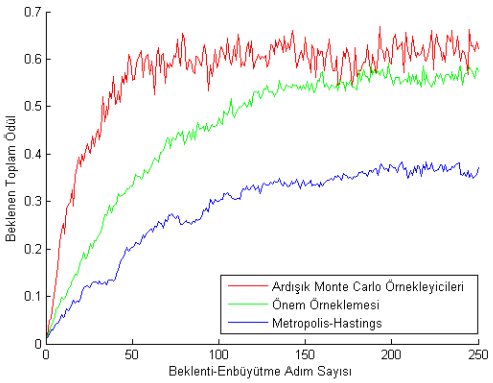
Ek olarak, denklem (12)'de sunulan üs fonksiyonu $\eta(n) > 1$ olan köprü fonksiyonları kullanıp iyice tavlayarak yakınsamayı hızlandırmayı hedefledik.

Önerdiğimiz yöntemimizi üssel çarpanları $\{0, 0.1, 0.33, 1, 3, 10\}$ olan köprü fonksiyonları için ve her köprü fonksiyonunu 100 örnekle kestirerek dağ-araba problemi üzerine uyguladık. Şekil 2'de gözüktüğü üzere, algoritmamız tekdüze rastgele ilke ile başladıktan sonra beklendiği üzere ilke ödül olasılığını yükseltecek şekilde yakınsamaktadır.

Bunun yanı sıra, kullandığımız BE algoritmasının beklenti-adımındaki beklenen değerleri sunduğumuz MH çekirdekli AMCÖlerin yanı sıra [6]'daki önem örneklemesi ile ve de AMCÖ kullanmadan türettiğimiz MH yöntemi ile kıyasladık. Yakınsama hızlarını daha belirgin olarak görebilmek için bütün algoritmaları çok az miktarda ve eşit sayıda örnek için çalıştırdık. Şekil 3'de görülen her yöntemi 10 kere çalıştırıp ortalama alınmış sonuçlara göre önerdiğimiz yöntem



Şekil 2: MH çekirdekli AMCÖ kullanan BE algoritmasıyla örneklenmiş durum gezingelerinin adım adım yakınsaması.



Şekil 3: MH çekirdekli AMCÖ'nün performasının önem örnekleyicisi ve MH ile karşılaştırmalı sonuçları.

kıyasladığımız yöntemlere göre aynı sayıda örnek için daha hızlı bir yakınsama sağladı.

V. VARGILAR

Türettiğimiz MH çekirdekli AMCÖ yöntemi başarılı bir şekilde beklenen değerleri hesaplamakta ve BE algoritmasının eniyi ilkeye yakınsamasını sağlamaktadır. Bunun yanı sıra, örnekleri de önem örneklemesine göre daha verimli kullanmakta ve yakınsama sürecini hızlandırmaktadır.

Ancak, çok az sayıda örnekle yakınsama performansını ölçtüğümüz deneyde, yöntemimizin performansının MH ile arasındaki farkın bu kadar büyük olmasının MH yönteminin az sayıda örnek kullanıldığı için tam olarak ısınamadan her BE yinelemesinde tekrar tekrar baştan çalışmasından kaynaklandığını düşünüyoruz.

Şekil 3'deki sonuçlarda yöntemimizin başarımı diğer yöntemlere göre yüksek bir varyansa sahip gibi gözükmektedir. Ancak bunun nedeni, her yöntem için aynı sayıda örnek kullanabilmek adına köprü fonksiyonu kestirimi başına düşen örnek sayısının azalması ve haliyle de kestirimin varyansının artmasıdır.

A. Gelecek Çalışmalar

Bu yaptığımız çalışmada sabit uzunluktaki durum-eylem gezingeleri üzerinde çalıştık. Ancak, önerdiğimiz bu yöntemin doğrudan böyle bir kısıtı mevcut değildir. Bu yüzden, ileri-geri çekirdeklerimizi MH çekirdekleri yerine tersinir atlama Markov zinciri Monte Carlo çekirdekleri seçerek yöntemimizi değişken T uzunluktaki $(x_{0:T}, a_{0:T})$ gezingeleri için de geliştirmeyi planlıyoruz.

Çalışmamızda, AMCÖleri BE algoritmasının beklenti adımında kullanarak durum-eylem gezingelerinin sonsal dağılımından örnekler çektik. Bunun yanı sıra, π ilkesini de çeşitli parametrelerle ifade edip genişletilmiş $(x_{0:T}, a_{0:T}, \pi)$ durum-eylem gezingeleri ve ilkeler olasılık uzayından örnekleme yaparak BE algoritmasına olan gerekliliği ortadan kaldırmayı hedefliyoruz.

Son olarak ise, modele dayalı PÖ problemi için önerdiğimiz bu yöntemi geliştirerek modelden bağımsız PÖ problemine de uygulanabilir hale getirmek ve bu koşullar altındaki performansını genel PÖ algoritmaları ile kıyaslayamak istiyoruz.

KAYNAKÇA

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control 3rd Edition, Vol. I*, ser. Athena Scientific optimization and computation series. Athena Scientific, 2007, vol. 2, no. 1.

[2] C. Szepesvári, *Algorithms for Reinforcement Learning*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010, vol. 4, no. 1.

[3] E. Alpaydin, *Introduction to Machine Learning*, T. Dietterich, C. Bishop, D. Heckerman, M. Jordan, and M. Kearns, Eds. The MIT Press, 2004, vol. 56, no. 2.

[4] M. Toussaint and A. Storkey, "Probabilistic inference for solving discrete and continuous state Markov Decision Processes," in *Proceedings of the 23rd International Conference on Machine Learning*. New York, New York, USA: ACM, 2006, pp. 945–952.

[5] T. Furmston and D. Barber, "Efficient Inference in Markov Control Problems," in *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*. AUAI Press, 2011, pp. 221–229.

[6] O. Sönmez and A. T. Cemgil, "Modele Dayalı Pekiştirme ile Öğrenme için Önem Örneklemesi (Importance Sampling for Model-Based Reinforcement Learning)," in *Proceedings of 20th IEEE Signal Processing ve Communication Applications Conference (SIU)*, 2012.

[7] M. Hoffman and A. Jasra, "Trans-dimensional MCMC for Bayesian Policy Learning," *Neural Information Processing Systems*, vol. 20, pp. 1–8, 2008.

[8] P. Del Moral and A. Doucet, "Sequential monte carlo samplers," *Journal of the Royal*, no. December, pp. 1–29, 2006.

[9] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction." *IEEE Transactions on Neural Networks*, vol. 9, no. 1, p. 1054, 1998.

[10] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, 1967.