KAPASİTE KISITLI STOKASTİK STOK PROBLEMİ İÇİN BENZETİMSEL DİNAMİK PROGRAMLAMA SEZGİSELİ

Mustafa ÇİMEN
Dr., Hacettepe Üniversitesi, İ.İ.B.F., İşletme Bölümü

Sedat BELBAĞ
Dr., Gazi Üniversitesi, İ.İ.B.F., İşletme Bölümü

Mehmet SOYSAL
Dr., Hacettepe Üniversitesi, İ.İ.B.F., İşletme Bölümü

ÖZET

Rekabetçi bir ortamda faaliyet gösteren işletmeler için belirsizlik altında stok yönetimi oldukça önemlidir. Bu çalışma kapasite kısıtlı stokastik stok problemine Benzetimsel Dinamik Programlama (Approximate Dynamic Programming-ADP) tabanlı bir sezgisel yöntem önermektedir. Önerilen sezgisel yöntem, Geçici Farklarla Öğrenme (Temporal Difference Learning-TD) yöntemini kullanmaktadır. Söz konusu problem için sezgisel yöntemden elde edilen çözümlerin performansı, Dinamik Programlama algoritmasından elde edilen optimal sonuçlar ile karşılaştırılarak değerlendirilmiştir. Sayısal örnekler sonucunda ADP'nin kısa süre içerisinde ümit vadeden sonuçlar verdiği gözlenmiştir.

Anahtar kelimeler: Stokastik Stok Problemi, Kapasite Kısıtı, Dinamik Programlama, Sezgisel, Benzetimsel Dinamik Programlama

AN APPROXIMATE DYNAMIC PROGRAMMING HEURISTIC FOR THE CAPACITATED STOCHASTIC INVENTORY PROBLEM

ABSTRACT

Inventory management under uncertainty is crucial for enterprises operating in competitive environment. This study presents a heuristic based on Approximate Dynamic Programming for the capacitated stochastic inventory problem. The proposed heuristic employs Temporal Difference Learning method. The performance of the heuristic solutions for the aforementioned problem has been assessed relative to the optimal solutions obtained through Dynamic Programming algorithm. Numerical experiments show that Approximate Dynamic Programming provides promising solutions within relatively short computational times.

Keywords: Stochastic Inventory Problem, Capacity Constraint, Dynamic Programming, Heuristic, Approximate Dynamic Programming

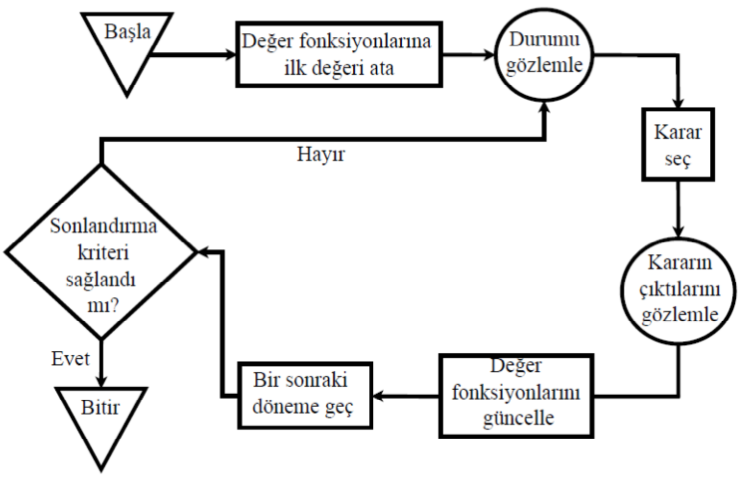
1. GİRİŞ

İşletmeyi oluşturan her departman bulundurulması gereken stok miktarı konusunda farklı yaklaşımlara sahiptir. Finans yöneticileri işletme kaynaklarını farklı yatırımlarda kullanabilmek için olabildiğince az stok bulundurulmasını amaçlarken, satın alma departmanı iskontolardan faydalanabilmek için olabildiğince fazla miktarda ürün almayı amaçlamaktadır. Stok yönetimi farklı amaçlara sahip departmanlar arasındaki dengeyi sağlamaya çalışır (Axsäter, 1990). Stok yönetimi bu dengeyi göz önünde bulundurarak taleplerin zamanında karşılanması için gereken optimal üretim ve stok miktarını belirleyebilmeyi amaçlar. İşletmeler başarılı stok yönetimleri sayesinde stok maliyetlerini düşürerek rakiplerine karşı rekabet avantajı sağlayabilirler. Bunun yanında başta talepte olmak üzere üretim teknolojileri, pazarlama ve lojistik gibi unsurlarda gerçekleşen çevresel değişimler de stok yönetiminin önemini arttırmaktadır.

Stok optimizasyonu problemleri çoğunlukla lineer programlama (LP), karma tamsayılı programlama (KTP) ve Markov karar süreci (MKS) gibi yöntemler kullanılarak modellenmiştir. MKS ile modellenen problemler, yaygın olarak dinamik programlama (DP) yöntemi kullanılarak çözülmektedir. DP, karar problemlerini aşamalara bölerek her bir aşamayı diğerlerinden bağımsız olarak çözer. Çözüm için her aşamada yer alan olası durumlar listelenir, her bir olası durum için olası tüm karar alternatifleri ve bu kararların beklenen değerleri (expected value) hesaplanır. Böylece DP karar problemlerinin karmaşık yapısını sadeleştirir ve belirli sayıda küçük karar problemini çözerek en iyi politikaya ulaşır.

Durum ve karar alternatifi sayısının sınırlı olduğu MKS problemlerinde DP genellikle diğer yöntemlere nazaran daha hızlı sonuç üretir. Ancak durum ve karar sayısının arttığı, özellikle çok boyutlu stokastik karar problemlerinde (örneğin her bir durumun birden fazla değişken tarafından tanımlandığı ya da her bir kararda birden fazla değişkenin değerinin belirlendiği problemlerde) bu performans hızla düşmektedir. Zira durum ve/veya karar sayısının artması, DP'nin ihtiyaç duyduğu hafızanın ve çözüm süresinin artmasına sebep olmaktadır. Problem büyüdükçe ihtiyaç duyulan zaman ve hafıza, karar vericinin sahip olduğu kaynakları aşabilir. Böyle durumlar ile baş edebilmek için optimale yakın sonuçlar elde edebilen sezgisel yöntemlere ihtiyaç duyulmaktadır. Bu sezgisel yöntemlerden biri de Benzetimsel Dinamik Programlama (Approximate Dynamic Programming; ADP) algoritmasıdır.

ADP algoritması DP'ye benzer şekilde karar problemlerini aşamalara bölerek her bir aşamanın maliyetini/getirisini (buradan itibaren "değer" olarak anılacaktır) ayrı ayrı hesaplar. Ancak bunu, DP algoritması gibi olası her durum için tüm karar alternatiflerinin beklenen değerlerini hesaplamak yerine, bir benzetimin (simülasyon) rassal sonuçları üzerinden yaklaşık bir beklenen değer hesaplar. Dolayısıyla rassal değerler üzerinden kararların kalitesini karşılaştırarak iyi bir karar bulmaya çalışır. Şekil 1'de genel bir ADP algoritması örnek olarak verilmiştir. Bu yaklaşım DP'ye kıyasla daha büyük boyutlu ve çözülmesi daha uzun zaman alan problemlerin çözülmesini mümkün kılar. Bunun yanında, ADP algoritması benzetim temelli yaklaşımı sayesinde rassal değişkenin dağılımının ya da geçiş fonksiyonlarının bilinmediği, dolayısıyla DP algoritmasında beklenen değerlerin hesaplanamadığı problemlerin çözümünde de kullanılabilir (Powell, 2007). ADP yöntemi talep tahmini (Iida ve Zipkin, 2006), kaynak/kapasite tahsisi (Erdelyi ve Topaloglu, 2011), rotalama (Yu ve Bertsekas, 2013) ve stok yönetimi (Pratikakis vd., 2010) gibi problemlerin çözümlerinde kullanılmaktadır.



Şekil 1: *Genel bir ADP algoritması. Algoritma durumu gözlemleme, karar verme, sonuç durumunu ve ödülü gözlemleme döngüsü simülasyon uzunluğu boyunca tekrar ederek değer fonksiyonlarını tahminlemeye çalışır***.**

Bu çalışma kapasite kısıtı ve kesikli stokastik talep varsayımlarıyla bir parti büyüklüğü problemini ele almaktadır. Problemde her dönem başında üretim siparişi verildiği ve siparişlerin teslim edilme sürelerinin gözardı edilebilecek kadar küçük olduğu varsayılmaktadır. Sabit sipariş maliyeti, her dönem sipariş verildiğinden gözardı edilmiştir. Çalışmanın amacı, söz konusu problemin MKS modeli için sonlu planlama ufkunda indirgenmiş stok maliyetlerini minimize eden optimal politikalara yakın politikalar üretebilen ADP tabanlı bir sezgisel yöntem geliştirilmesidir.

Çalışmanın bundan sonraki bölümleri şu şekilde yapılandırılmıştır. İkinci bölümde kapasite kısıtlı stok optimizasyonu problemine ve stok yönetiminde ADP kullanımına ilişkin bir literatür taraması yapılmıştır. Üçüncü bölümde ele alınan problemin varsayımları ve MKS modeli sunulmaktadır. Dördüncü bölümde geliştirilen ADP tabanlı sezgisel yöntem açıklanacaktır. Beşinci bölümde sezgisel yöntemin performansının değerlendirilmesi amacıyla sayısal örnekler sunulacaktır. Bu amaçla oluşturulan örnek problemler için DP metoduyla elde edilen optimal politikalar, sezgisel yöntemin ürettiği politikalarla indirgenmiş stok maliyetleri açısından karşılaştırılacaktır. Son bölümde ise elde edilen sonuçlar tartışılmış ve gelecek çalışmalara yönelik öneriler sunulmuştur.

2. LİTERATÜR TARAMASI

Scarf (1959)'ın çalışması belirsizliği dikkate alarak stok optimizasyonu yapan en önemli çalışmalardan birisidir. Bu çalışma, tek bir ürünün üretildiği üretim sistemi için optimal stok politikalarını ortaya koymuştur. Politika stok miktarının *s* ile tanımlanan minimum stok seviyesine düştüğünde *S* ile tanımlanan stok seviyesi kadar yükseltilmesini sağlamaktadır ve literatürde (*s, S*) politikası olarak anılmaktadır. Sonrasında Scarf (1959)'ın çalışmasındaki çeşitli varsayımlar esnetilerek farklı stok optimizasyonu problemleri çalışılmıştır (örneğin, stokastik temin zamanı varsayımı, Ehrhardt, 1984; sonsuz planlama ufku, Iglehart, 1963; Markov talep varsayımı, Sethi ve Cheng, 1997; hazırlık süresi varsayımı, Sobel ve Zhang, 2001).

Scarf'ın çalışması ve bu çalışmayı takiben yukarıda anılan diğer yayınlar (*s, S*) politikasının pek çok varsayım altında optimal politika olduğunu ortaya koymuştur. Ancak kapasite kısıtı bu varsayımlar arasında değildir. İşletmelerde gerek tesislerde gerekse depolarda sınırlı miktarda ürün bulundurulması gerekebilir. Kapasite kısıtı, üretilmesi ve/veya elde bulundurulması gereken ürün miktarını etkilemektedir. Gelecekte karşılaşılması muhtemel maliyetler bugün verilen kararlara bağlı olduğu için kapasite sınırını dikkate alan problemlerin çözümü bu sınırı göz ardı eden problemlerin çözümüne göre daha zordur (Levi vd., 2008). Kapasite kısıtlı stok problemlerinin, karmaşıklık seviyesi arttıkça çözüm süresi de üssel olarak artan problem (NP-hard problem) yapısında olduğu belirtilmiştir (Shaw ve Wagelmans, 1998).

Tablo, literatürdeki kapasite kısıtlı stokastik stok problemine yönelik çalışmaların özetini sunmaktadır. Literatürdeki çalışmalarda, kapasite kısıtlı stokastik stok problemlerine yönelik matematiksel ispat ya da DP kullanılarak optimal politikalar üretilmiştir. DP sonlu planlama varsayımı altındaki kapasite kısıtlı stokastik stok problemlerinde sıklıkla kullanılmasına rağmen, ihtiyaç duyduğu hafıza ve çözüm süresinin uzunluğundan dolayı problem boyutunun büyümesi DP algoritmasının performansını olumsuz yönde etkilemektedir. Diğer yandan, DP'ye alternatif olabilen ADP algoritması kapasite kısıtı olmayan stokastik stok problemlerinde kullanılmış olsa da, bilgimiz dahilinde kapasite kısıtlı stokastik stok problemlerinde henüz kullanılmamıştır. Bu çalışma literatürdeki çalışmalardan farklı olarak sonlu planlama ufku varsayımı altındaki kapasite kısıtlı stokastik stok probleminin çözümü için bir ADP algoritması önermektedir.

Tablo 1: Kapasite kısıtlı stokastik stok problemine ilişkin çalışmalar. Listelenen tüm çalışmalar kesikli talep ve periyodik sipariş varsayımlarını kullanmaktadır.

Çalışmalar	Kapasite kısıtı	MKS ile modelle	Sonlu planlama	Sezgisel yöntem	Model türü
Federgruen ve Zipkin (1986a,b)	✓	-	-	-	Matematiksel
Kapuściński ve Tayur (1998)	✓	-	✓	-	Simülasyon
Roundy ve Muckstadt (2000)	✓	-	-	Talep Dağılımı Yakınsaması	Matematiksel
Gallego ve Scheller-Wolf (2000)	✓	-	✓	-	Matematiksel
Levi vd. (2008)	✓	-	✓	Yarı Miyopik Politikalar	Matematiksel
Chen vd. (1994)	✓	-	✓	-	DP
Aviv ve Federgruen (1997)	✓	✓	✓	-	DP
Shaw ve Wagelmans (1998)	✓	-	✓	-	DP
Van Hoesel ve Wagelmans (2001)	✓	-	✓	Tam Polinomsal Yakınsama	DP
De Vericourt vd. (2002)	✓	✓	-	-	DP
Özer ve Wei (2004)	✓	-	✓	-	DP
Shervais vd. (2003)	-	-	✓	Uyarlamalı Kritik Öğrenme	ADP
Topaloglu ve Kunnumkal (2006)	-	✓	✓	Lagrange Gevşetme Temelli Yakınsama	ADP
Katanyukul vd. (2011)	-	✓	-	Sarsa, Rollout, Hindsight	ADP
He (2013)	-	-	✓	Lagrange Yöntemi	ADP
Bu çalışma	✓	✓	✓	Geçici Farklarla Öğrenme	ADP

3. KAPASİTE KISITLI STOKASTİK STOK PROBLEMİ

Bu çalışma sonsuz planlama ufku için kapasite kısıtı varsayımı altındaki bir stokastik stok problemine uygun çözüm değerini bulmayı amaçlamaktadır. Ele alınan problemin periyodik karar yapısı sebebiyle problem MKS olarak modellenmiştir. Çalışmanın amacı, oluşturulan bu MKS modeli için indirgenmiş toplam stok maliyetlerini minimize edecek politikaların hesaplamasıdır.

Varsayılan stok sisteminde periyodik olarak her dönem başında mevcut stok durumu kontrol edilerek, o dönemde üretilecek ürün miktarı belirlenir. Sistemi açıklayan MKS modelinde durumlar her *t* dönemi için dönem başındaki stok miktarı (i_t) olarak tanımlanmıştır. Karar vericinin t dönemi başında verebileceği tüm olası üretim miktarı kararları (q_t) modelin karar uzayını oluşturur. Üretim süreleri gözardı edilerek sipariş edilen ürünlerin sipariş anında stoklara ulaştığı varsayılmıştır. Ancak üretim miktarları, üretim tesisinin kapasitesiyle (c) sınırlıdır. Ek olarak dönem sonunda elde kalan ve bir sonraki dönemde depolanacak ürünler için de bir depo kapasitesi (i^{max}) tanımlanmıştır. Dönem sonunda elde kalan ürünler stok kapasitesini aştığında kapasite fazlası miktarın maliyetsiz olarak imha edildiği varsayılmıştır.

Stok seviyeleri gerçekleşen üretim ve talebe göre değişir. Herhangi bir t döneminde dönem başı stok seviyesi (i_t), herhangi bir a kararının tercih edilmesi ile karar sonrası stok seviyesine (i_t^a) şu şekilde dönüşür:

$$i_t^a = i_t + q_t \tag{1}$$

Karar sonrasında sistemin durumu, talebi karşılamak için kullanılabilecek stok miktarını gösterir. Talebin (D_t) gözlenmesiyle stoklardaki ürünler talebi karşılamak için kullanılır. Dönem sonunda elde kalan ürünler, depo kapasitesi de dikkate alınarak, bir sonraki dönem için dönem başı stok miktarını belirleyecektir:

$$i_{t+1} = \min(i^{max}, i_t^a - D_t) \tag{2}$$

Dönem sonu stok miktarı, tesis kapasitesini aşamayacağı için maksimum depolama kapasitesi ve elde kalan stok miktarı arasından en düşük olan olacaktır. $\Gamma_t(d)$, t dönemindeki talebin Poisson olasılık dağılımını göstermek üzere, beklenen dönem sonu elde kalan stok miktarı (i_t^+) ve beklenen karşılanamayan talep (i_t^-) sırasıyla şu şekilde yazılabilir:

$$i_t^+ = \sum_{d=0}^{i_t^a} \Gamma_t(d)\,(i_t^a - d) \tag{3}$$

$$i_t^- = \sum_{d=i_t^a}^{\infty} \Gamma_t(d)(d - i_t^a) \tag{4}$$

Talebin mevcut stoklarla karşılanamadığı durumlarda sonraki dönemlere ertelendiği varsayılmıştır. Bu durumda toplam maliyete, ertelenen talep miktarı ile doğru orantılı bir ceza maliyeti (Θ) eklenir. Her dönemin sonunda elde kalan stoklar ise, elde kalan stok miktarı ile doğru orantılı bir elde bulundurma maliyetine (H) sebep olur. Bir durumdan başka bir duruma geçerken elde edilen beklenen ödül[1] (*R*), üretim, elde bulundurma ve ceza maliyetlerinin toplamından oluşur. Üretim, elde bulundurma, ceza ve maliyetleri şu şekilde tanımlanabilir:

$$U_t = uq_t \tag{5}$$

$$H_t = hi_t^+ \tag{6}$$

$$\Theta_t = \theta i_t^- \tag{7}$$

Burada u birim üretim maliyetini, h birim elde bulundurma maliyetini ve θ birim ceza (stoksuz kalma) maliyetini göstermektedir. Sonuç olarak bir *s* başlangıç durumunda herhangi bir *a* kararı tercih edilerek ortaya çıkacak beklenen ödül bu üç maliyetinin toplamına eşit olacaktır.

$$R_{st}^a = U_t + H_t + \Theta_t \tag{8}$$

Bu problemin optimal çözümü aşağıdaki denklemi minimize eden üretim kararlarının bulunması ile elde edilir.

[1]Beklenen ödül, problemin yapısına göre maliyet ya da gelir olabilir.

$$V_t(s) = \min_a \{R_{st}^a + \gamma R_{st+1}^a + \gamma^2 R_{st+2}^a + \gamma^3 R_{st+3}^a + \cdots + \gamma^{T-t} R_{sT}^a\} \quad (9)$$

Yukarıdaki denklemde gelecek ödüller, finans teorisine uygun şekilde bir iskonto oranıyla (γ) bugünkü değerlerine indirgenmiştir. $V_t(s)$, t döneminde s durumunda bulunmanın beklenen değeri, bir başka deyişle s durumunun değer fonksiyonudur. Geleneksel DP yönteminde bu eşitlik, Bellman Denklemi'nin son dönemden geriye doğru tekrarlanarak çözülmesiyle optimize edilir. Herhangi bir s durumundan a kararını tercih ederek s' durumuna $\beta_{ss'}^a$ olasılığı ile geçmesi sonucunda s durumunun değer fonksiyonu Bellman denklemi olarak şu şekilde yazılabilir.

$$V_t(s) \leftarrow \min_a \left\{ R_s^a + \gamma \sum_{s'} \beta_{ss'}^a V(s') \right\} \quad (10)$$

Burada $V(s')$, s' durumunun değer fonksiyonunu göstermektedir. DP küçük boyutlu stokastik stok problemlerini optimal olarak çözebilmesine rağmen, kesikli olasılık dağılımına sahip durum sayısının artması ile boyutu büyüyen problemleri optimal olarak çözemez. Bu sebeple, optimal sonuç olmasa da optimale oldukça yakın sonuçlar elde edilebilen sezgisel yöntemlere başvurulur. Bu çalışmada ADP yaklaşımı kullanılarak geliştirilen bir sezgisel yaklaşım sunulacaktır.

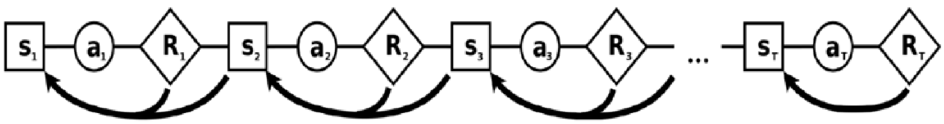
4. SEZGİSEL ADP ALGORİTMASI

Tipik bir ADP algoritması, gözlem ve iyileştirme aşamalarından oluşur. Gözlem aşamasında mevcut karar politikasının sonuçları gözlenir ve bu politika için değer fonksiyonu tahminleri belirlenir. Ancak değer fonksiyonlarını belirlemek için DP'nin yaptığı şekilde beklenen değerlerini hesaplamak yerine, ADP algoritmasında politikanın uygulandığı bir simülasyonda gözlenen rassal sonuçlar üzerinden değer fonksiyonları tahminlenmeye çalışılır. Daha sonra iyileştirme aşamasında bu tahminler kullanılarak mevcut karar politikası, varsa daha iyi bir politikayla değiştirilir. Böylece DP'deki gibi her durum için tüm olası kararları, her bir kararın olası sonuçlarını ve sonuçların olasılıklarını içeren karmaşık ve uzun bir hesaplama yapılmaz. Bunun yerine, kararların, sonuçların ve olasılıkların sadece simülasyonda gözlenen kısmı kullanılarak değer fonksiyonu tahminlemesi yapılır. Bu tahminleme için ADP yaklaşımında genellikle bir ödüllü öğrenme yöntemi olan Geçici Farklarla Öğrenme (Temporal Difference Learning; TD) yöntemi kullanılmaktadır.

TD yöntemi değer fonksiyonlarının mevcut değerleriyle simülasyonda üretilen rassal değerler arasındaki farkları kullanır. Simülasyonda gözlenen her ödül, değer fonksiyonunun alabileceği olası bir değerdir. TD algoritması ödülü gözlediğinde o ödülün alındığı durumun değer fonksiyonunu ödülün değerine yaklaştırır. Bu şekilde gözlenen her ödülle yukarıya ya da aşağıya çekilen değer fonksiyonu, simülasyon yeteri kadar tekrarlandığında ortalama bir değere yaklaşacaktır.

TD yönteminde öncelikle içinde bulunulan durum için bir değer gözlemi yapılmalıdır. DP, bu değer için Bellman denklemini (10) çözer. TD ise simülasyonda elde edilen tek bir rassal ödülü (R_t) ve sistemin bir sonraki durumunu (s') gözlemleyerek bir değer fonksiyonu örneği (v_s) hesaplar (bkz. Şekil 2):

$$v_s = R_t + \gamma V(s') \tag{11}$$



Şekil 2: Tek Adımlı TD güncellemesi. s simülasyonun ziyaret ettiği durumları, a verilen kararları, R ise kararların gözlenen ödüllerini göstermektedir.

Hesaplanan örnek değerler üzerinden değer tahmini yapmak, simülasyon yönteminin de temelini oluşturur. TD ile geleneksel simülasyon yöntemi arasındaki en önemli fark TD yönteminin bu değer örneklerini gözlemlendikten hemen sonra kullanması ve karar politikasının bu doğrultuda değiştirilmesidir. v_s değerinin ve s durumunun değer fonksiyonunun mevcut değerinin bir ağırlıklı ortalaması alınarak değer fonksiyonu güncellenir. α yeni gözlemin ağırlığı olmak üzere yeni tahmînî değer fonksiyonu ($\hat{V}(s)$) şu şekilde hesaplanır:

$$\hat{V}(s) \leftarrow \alpha v_s + (1-\alpha)\hat{V}(s) \tag{12}$$

Böylece gözlemler, geçmiş bilgi ve güncel tahminleri kullanarak değer fonksiyonlarını anlık olarak değiştirir. Değiştirilen değer fonksiyonu tahminleri, politikaların iyileştirilmesinde kullanılır:

$$\hat{a} \leftarrow \arg\min_{a} \left\{ \overline{R^{a}} + \gamma \sum_{s'} \beta_{ss'}^{a} \, \hat{V}_{t+1}(s') \right\} \tag{13}$$

TD yöntemi tahminlerin iyileştirilmesi için örneklem kullanması bakımından geleneksel simülasyon yöntemine benzemektedir. Diğer yandan, TD yönteminde değer fonksiyonu tahminlerinin simülasyonun geri kalanında gözlenecek ödülleri temsil etmesi, DP algoritması ile benzerlik göstermektedir. Özetle, TD yönteminin hem geleneksel simülasyon yönteminin hem de DP algoritmasının faydalı özelliklerini kullandığı ve bu iki yöntemin arasında yer aldığı söylenebilir (Sutton ve Barto, 1998). TD yöntemine ait genel bir algoritma Şekil 3'te görülmektedir.

Aşama 1: Başlangıç

Tüm $s \in S$ için değer fonksiyonlarına ($\hat{V}_t(s)$) ilk değer ata
Bir s başlangıç durumu seç

i = 1'den *İ*'ye kadar

$t = 1$'den T'ye kadar

Aşama 2: Karar Seçimi

Tüm $a \in A$ için beklenen ödülü ($\overline{R^{a}}$) hesapla
$\hat{a} \leftarrow \arg\min_{a}\{\overline{R^{a}} + \gamma \sum_{s'} \beta_{ss'}^{a} \hat{V}_{t+1}(s')\}$

Aşama 3: Değer Fonksiyonlarının Güncellenmesi

Bir sonraki durumu (s') ve gerçekleşen ödülü (R) gözlemle
$v_s = R + \gamma \hat{V}_{t+1}(s')$
$\hat{V}_t(s) \leftarrow \alpha v_s + (1-\alpha)\hat{V}_t(s)$

Aşama 4: Bir Sonraki Yinelemeye Git

$s \leftarrow s'$

Döngüyü Bitir

Döngüyü Bitir

Şekil 2: Genel bir TD algoritması. İ, ADP simülasyonunun uzunluğunu göstermektedir.

Bu yaklaşım kullanılarak kapasite kısıtı altında stokastik parti büyüklüğü problemi için bir sezgisel yöntem geliştirilmiştir. Geliştirilen yöntem, DP gibi sonlu planlama ufkuna sahip karar problemini aşamalara bölerek son dönemden ilk döneme doğru çözmektedir. Ancak değer fonksiyonlarının güncellenmesi için **tüm** durumların **beklenen** değerleri yerine, sadece **seçilen bir grup** durumun **tahmînî beklenen** değeri hesaplanmaktadır. Bir s durumunun değer fonksiyonu tahmini, tüm olası sonuçların değerinin göz önüne alınması yerine, s durumunun başlangıç durumu olduğu bir simülasyonun n kez çalıştırılmasıyla elde edilir. Son olarak, tüm durumların değer fonksiyonlarına simülasyon çalıştırılmadan önce sezgisel bir **ilk değer** atanmaktadır. Bu sezgisel ilk değerler her durumun beklenen elde bulundurma ve ceza maliyetleri kullanılarak aşağıdaki şekilde hesaplanır:

$$\hat{V}_t(s) = \gamma^{t-1}(U_t + H_t) \tag{14}$$

Algoritma, sondan başa doğru tüm dönemlerde seçilen tüm durumlar için n kez çalıştıktan sonra sonlandırılır. Durumlar için kaydedilen tahmînî değer fonksiyonları, sezgisel kararların verilmesi için saklanır. Herhangi bir durum için karar verilmesi gerektiğinde bu tahmînî değer fonksiyonları Bellman denklemine benzer şekilde kullanılarak karar belirlenecektir:

$$\hat{a}_t \leftarrow \arg\min_a \left\{R_s^a + \gamma \textstyle\sum_{s'} \beta_{ss'}^a \hat{V}_{t+1}(s')\right\} \tag{15}$$

Şekil 4, geliştirilen sezgisel ADP yaklaşımı için örnek bir algoritmayı göstermektedir.

Aşama 1: Başlangıç

Tüm $s \in S$ ve $t \in T$ için değer fonksiyonlarına ($\hat{V}_t(s)$) ilk değer ata:

$$\hat{V}_t(s) = \gamma^{t-1}(U_t + H_t)$$

Bir s başlangıç durumu seç

$t = T$'den 1'e kadar

Tüm $s \in N$ için

i = 1'den İ'ye kadar

$t' = t$ 'den T'ye kadar

Aşama 2: Karar Seçimi

Tüm $a \in A$ için beklenen ödülü ($\overline{R^a}$) hesapla

$\hat{a} \leftarrow \arg\min_a \left\{ \overline{R^a} + \gamma \sum_{s'} \beta^a_{ss'} \hat{V}_{t+1}(s') \right\}$

Aşama 3: Değer Fonksiyonlarının Güncellenmesi

Bir sonraki durumu (s') ve gerçekleşen ödülü (R) gözlemle

$v_s = R + \gamma \hat{V}_{t+1}(s')$

$\hat{V}_t(s) \leftarrow \alpha v_s + (1 - \alpha)\hat{V}_t(s)$

Aşama 4: Bir Sonraki Yinelemeye Git

$s \leftarrow s'$

Döngüyü Bitir

Döngüyü Bitir

Döngüyü Bitir

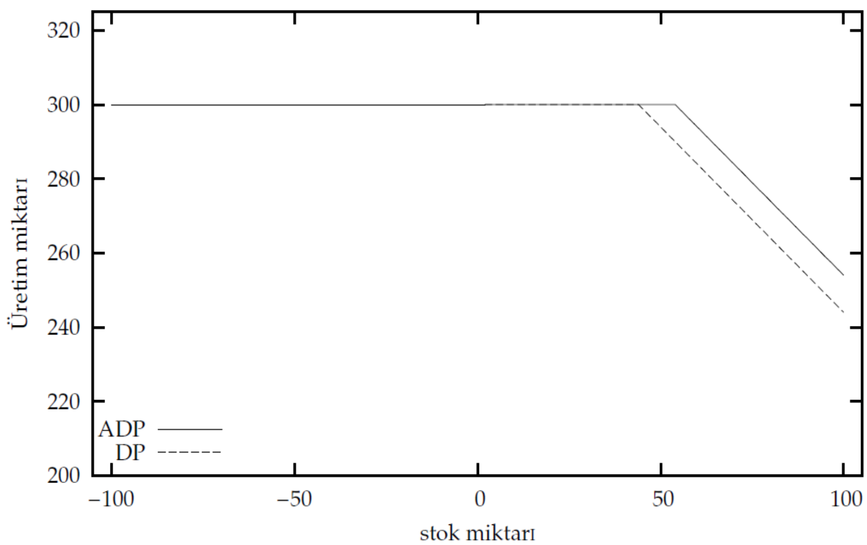
Döngüyü Bitir

Şekil 4: Kapasite kısıtı ve sonlu planlama ufku altında parti büyüklüğü problemi için geliştirilmiş sezgisel ADP algoritması. N değer fonksiyonu güncellenen durumları, İ ise ADP simülasyonunun uzunluğunu göstermektedir.

5. SAYISAL ÖRNEKLER

Bu bölümde geliştirilen ADP tabanlı sezgisel yöntemin performansı ile ilgili örnekler sunulacaktır. Öncelikle rassal olarak seçilen bir örnek problem için DP ve ADP politikaları karşılaştırılacaktır. Ardından, ADP'nin farklı problem parametreleri için de performansının göz önüne serilmesi amacıyla farklı stoksuz kalma maliyeti, üretim ve depolama kapasiteleri ve talep desenleri için DP ve ADP algoritmalarıyla elde edilen politikalar, bu politikaların ürettikleri indirgenmiş toplam stok maliyetleri açısından değerlendirilecektir.

Örnek problemde birim üretim maliyetinin (u) 1 ve elde bulundurma maliyetinin (h) 1, ceza maliyetinin (θ) 7, üretim kapasitesinin 300 birim ve depolama kapasitesinin 100 birim olduğu varsayılmaktadır. Üretilen ürün için her dönem 300 ortalamalı Poisson dağılıma sahip bir talep geldiği varsayılmaktadır. Planlama ufku 12 dönemdir. Bu problemin ilk dönemi için DP ve ADP algoritmalarının, başlangıç durumu -100 ile 100 arasında olduğunda izlediği politikalar Şekil 5'te gösterilmektedir.



Şekil 5: DP ve ADP politikalarının örnek problemin ilk dönemi için üretim miktarı kararları.

Şekil 5'ten anlaşılacağı üzere ADP ve DP algoritmaları problem için benzer şekilde eşik (stok hedefi, order up to level) politikasını benimsemekte, eşikler arasında küçük farklılıklar olmakla beraber (DP için eşik 354, ADP için 344'dür) yakın stok hedeflerine ulaşılmaya çalışılmaktadır. Şekilde ADP algoritmasının optimal politikadan biraz daha fazla ürün stoklamayı hedeflediği görülmektedir, ancak aradaki fark optimal eşiğin %3'ünden azdır. Diğer dönemlerdeki politikalarda da stok hedefleri arasında ortalama %5,5'lik bir fark vardır. Başlangıç stoğunun -100 ile 100 arasında olduğu durumlar için tüm dönemlerde üretim kararları arasındaki fark ise %2'nin altındadır. Hem politikaların benzerliği, hem stok hedefleri arasındaki farkın düşüklüğü göz önüne alındığında sezgisel ADP algoritmasının bu problem için ümit vadeden bir çözüm algoritması olduğu söylenebilir.

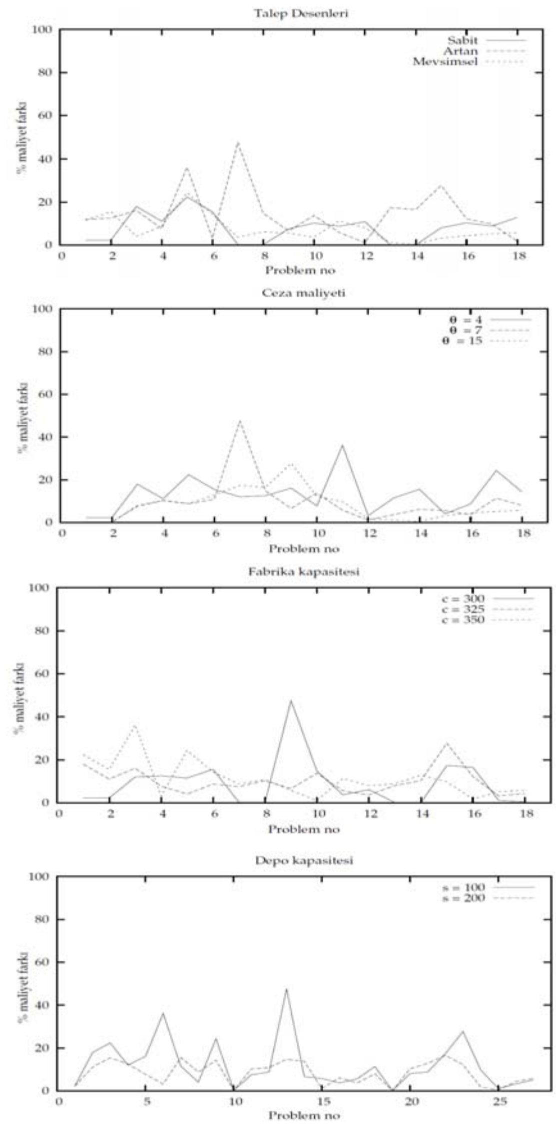
Ele alınan örnek, ADP politikasının optimal DP politikasına oldukça yakın olduğunu ortaya koymaktadır. Ancak problem parametreleri değiştikçe ADP yönteminin optimal politikaya kıyasla performansı değişebilir. ADP algoritmasının performansının indirgenmiş toplam maliyetler açısından incelenmesi ve bu performansın farklı maliyet, kapasite ve talep parametreleri için nasıl değiştiğinin ortaya konması amacıyla çeşitli ceza maliyetleri, üretim ve depo kapasiteleri ve talep desenlerinin farklı kombinasyonları için bir örnek seti oluşturulmuştur. Oluşturulan örnek seti için DP ve ADP politikaları elde edilmiştir. DP algoritması, oluşturulan problemleri yaklaşık 4 saatte çözerken ADP algoritması 35-40 dakika arasında çalıştırılmıştır. Ardından bu politikaların ürettiği stok maliyetleri her örnek için 1000 tekrarlı Monte Carlo simülasyonu yöntemi kullanılarak hesaplanmıştır. Tablo 2 örnek setinde kullanılan parametreleri, Tablo 3 ise kullanılan 3 farklı talep deseninde her dönem için ortalama talep miktarını göstermektedir. Şekil 6'da ADP politikalarının oluşturulan 54 problem için ürettiği maliyetler, her bir ceza maliyeti, stok ve üretim kapasitesi ve talep deseni seçeneği için ayrı ayrı değerlendirilmektedir.

Tablo 2: Örnek seti için kullanılan parametreler.

Parametreler	
Üretim Kapasitesi	300-325-350
Depo Kapasitesi	100-200
Talep	Sabit-Artan-Mevsimsel
Üretim Maliyeti	1
Elde Bulundurma Maliyeti	1
Ceza Maliyeti	4-7-15

Tablo 3: Örnek setinde kullanılan talep desenleri.

Dönem	Sabit Talep	Artan Talep	Mevsimsel Talep
1	300	123	368
2	300	150	329
3	300	197	260
4	300	225	218
5	300	232	259
6	300	258	336
7	300	306	378
8	300	347	304
9	300	369	267
10	300	381	242
11	300	393	296
12	300	395	330



Şekil 6: ADP politikalarının örnek problemler için ürettiği stok maliyetlerinin, optimal politikalara kıyasla yüzdelik farkı.

Şekil 6'dan görüleceği üzere ele alınan 54 problem için ADP ve DP politikalarının üretim maliyetleri arasındaki fark, ortalama yaklaşık %10 civarındadır. Ancak ADP algoritmaları hesaplama zamanlarında %85'e yakın bir tasarruf sağlamaktadır. Bu durum daha hızlı karar alınmasını sağlayacağı gibi, daha büyük ve DP'nin kullanılamadığı problemler için ADP'nin kullanılabilirliği açısından da ümit vadetmektedir.

Problem parametrelerinin ADP algoritmasının performansına etkisi incelendiğinde düşük stok kapasitesi ve artan talep deseni varsayımlarının ADP politikasının başarısını sınırladığı gözlenebilir. Ancak bu durumlar dışında problem parametrelerinin genelinde sezgisel ADP algoritması optimal politikaya yakın ve düşük maliyetli politikalar üretmeyi başarmıştır.

Sonuç olarak sezgisel ADP algoritmasının DP algoritmasından çok daha kısa sürede DP politikalarına yakın sonuçlar ürettiği söylenebilir. Bu sonuçlar, sezgisel ADP algoritmasının özellikle DP algoritmasının kullanılamadığı büyük problemlerde kullanılmasını teşvik etmektedir.

6. SONUÇLAR

Bu çalışmada kapasite kısıtı varsayımı altında stokastik talepli üretim sistemleri için parti büyüklüğü belirleme problemi ele alınmıştır. Seçilen problem için literatürde önerilen çözüm yöntemleri bulunsa da, problemin MKS modeli için yaygınlıkla önerilen DP yaklaşımı, olası durum ve karar sayısı arttığında kullanışsız hale gelmektedir. Bu çalışmada söz konusu MKS modelinin çözümü için ADP tabanlı bir çözüm yöntemi önerilmektedir. Geliştirilen ADP algoritması, değer fonksiyonlarını hızlı güncellemesi, durum uzayının yalnız bir bölümünü inceleyerek tahminde bulunması ve beklenen değer yerine simülasyon sonuçlarını kullanması gibi sebeplerle DP yöntemine göre çok daha hızlı sonuç vermekte ve DP'nin kullanılamadığı büyük boyutlu problemlerde de kullanılabilmektedir.

Çalışmada yer alan sayısal örnekler, ADP algoritmasının büyük boyutlu problemlerde kullanılmasını teşvik etmektedir. Seçilen ilk örnek problemde ADP politikasıyla DP politikası arasındaki ortalama üretim kararları farkının ortalama %2'nin altında olduğu görülmüştür. Ek olarak, her iki algoritma için Şekil 5'de gösterilen politikaların, birbirine yakın bir desene sahip olduğu görülmektedir. Dolayısıyla ADP politikalarının kısa süre içerisinde optimale yakın sonuçlar ürettiği sonucuna ulaşılabilir. Yürütülen ikinci sayısal çalışma, ADP algoritmasının farklı maliyet, kapasite ve talep durumları için ürettiği stok maliyetlerini optimal maliyetlerle karşılaştırmıştır. Bu örneklerde de ADP algoritmasının DP politikasından çok daha kısa sürede optimale yakın maliyetler ürettiği anlaşılmaktadır (bkz. Şekil 6). Sonuç olarak yürütülen sayısal örnekler, ADP algoritmasının DP'nin kullanılamadığı büyük boyutlu problemlerde kullanılabilecek, hızlı ve etkin bir sezgisel yaklaşım olduğunu göstermektedir.

Bu çalışmanın kısıtları arasında sayısal örneklerin boyutu gösterilebilir. Farklı talep desenlerinin, pozitif tedarik sürelerinin ve stokastik üretim miktarlarının ADP algoritmasının performansı üzerindeki etkilerinin ölçülmesi literatüre önemli bir katkı sağlayacaktır. Ek olarak bu çalışmada sunulan ADP algoritması, birden fazla ürünün üretildiği çok boyutlu problemler için de geliştirilerek kullanılabilir. Son olarak, ADP algoritması için Q-Learning, Sarsa gibi yöntemler denenerek daha hızlı ve etkin algoritmaların elde edilmesi gelecek çalışmalar için olası bir yönelim olabilir.

KAYNAKÇA

Aviv, Y., Federgruen, A. (1997). Stochastic Inventory Models With Limited Production Capacity and Periodically Varying Parameters, *Probability in the Engineering and Informational Sciences*, 11(1), 107-135.

Axsäter, S. (1990). Simple Solution Procedures For A Class Of Two-Echelon Inventory Problems, *Operations Research*, 38(1), 64-69.

Chen, H. D., Hearn, D. W., Lee, C. Y. (1994). A New Dynamic Programming Algorithm For The Single Item Capacitated Dynamic Lot Size Model, *Journal of Global Optimization*, 4(3), 285-300.

De Vericourt, F., Karaesmen, F., Dallery, Y. (2002). Optimal Stock Allocation For A Capacitated Supply System, *Management Science*, 48(11), 1486-1501.

Ehrhardt, R. (1984). (s, S) Policies For A Dynamic Inventory Model With Stochastic Lead Times, *Operations Research*, 32(1), 121-132.

Erdelyi, A., Topaloglu, H. (2011). Using Decomposition Methods to Solve Pricing Problems in Network Revenue Management, *Journal of Revenue & Pricing Management*, 10(4), 325-343.

Federgruen, A., Zipkin, P. (1986a). An Inventory Model With Limited Production Capacity and Uncertain Demands I. The Average-Cost Criterion, *Mathematics of Operations Research*, 11(2), 193-207.

Federgruen, A., Zipkin, P. (1986b). An Inventory Model With Limited Production Capacity And Uncertain Demands II. The Discounted-Cost Criterion, *Mathematics of Operations Research*, 11(2), 208-215.

Gallego, G., Scheller-Wolf, A. (2000). Capacitated Inventory Problems With Fixed Order Costs: Some Optimal Policy Structure, *European Journal of Operational Research*, 126(3), 603-613.

He, H. Z. (2013). An Approximate Dynamic Programming Approach For Computing Base Stock Levels. In The 19th International Conference on Industrial Engineering and Engineering Management (749-757). Springer Berlin Heidelberg.

Iglehart, D. L. (1963). Optimality of (s, S) Policies in the Infinite Horizon Dynamic Inventory Problem, *Management science*, 9(2), 259-267.

Iida, T., Zipkin, P. H. (2006). Approximate Solutions of A Dynamic Forecast-Inventory Model, *Manufacturing & Service Operations Management*, 8(4), 407-425.

Kapuściński, R., Tayur, S. (1998). A Capacitated Production-Inventory Model With Periodic Demand, *Operations Research*, 46(6), 899-911.

Katanyukul, T., Duff, W. S., Chong, E. K. (2011). Approximate Dynamic Programming For An Inventory Problem: Empirical Comparison, *Computers & Industrial Engineering*, 60(4), 719-743.

Levi, R., Roundy, R. O., Shmoys, D. B., Truong, V. A. (2008). Approximation Algorithms For Capacitated Stochastic Inventory Control Models, *Operations Research*, 56(5), 1184-1199.

Özer, Ö., Wei, W. (2004). Inventory Control With Limited Capacity and Advance Demand Information, *Operations Research*, 52(6), 988-1000.

Powell, W. B. (2007). Approximate Dynamic Programming: Solving The Curses of Dimensionality, John Wiley & Sons.

Pratikakis, N. E., Realff, M. J., Lee, J. H. (2010). Strategic Capacity Decision-Making in A Stochastic Manufacturing Environment Using Real-Time Approximate Dynamic Programming, *Naval Research Logistics (NRL)*, 57(3), 211-224.

Roundy, R. O., Muckstadt, J. A. (2000). Heuristic Computation of Periodic-Review Base Stock Inventory Policies, *Management Science*, 46(1), 104-109.

Scarf, H. (1959). The Optimality of (s, S) Policies in the Dynamic Inventory Problem, in K. J. Arrow, S. Karlin, P. Suppes (eds.), *Mathematical Methods in the Social Science*, Stanford University Press, Stanford.

Sethi, S. P., Cheng, F. (1997). Optimality of (s, S) Policies in Inventory Models With Markovian Demand, *Operations Research*, 45(6), 931-939.

Shaw, D. X., Wagelmans, A. P. (1998). An Algorithm For Single-Item Capacitated Economic Lot Sizing With Piecewise Linear Production Costs and General Holding Costs, *Management Science*, 44(6), 831-838.

Shervais, S., Shannon, T. T., Lendaris, G. G. (2003). Intelligent Supply Chain Management Using Adaptive Critic Learning, *Systems, Man and Cybernetics, Part A: Systems and Humans*, *IEEE Transactions on*, 33(2), 235-244.

Sobel, M. J., Zhang, R. Q. (2001). Inventory Policies For Systems With Stochastic and Deterministic Demand, *Operations research*, 49(1), 157-162.

Sutton, R. S., Barto, A. G. (1998). Reinforcement Learning: An Introduction, MIT Press, Masachusetts.

Topaloglu, H., Kunnumkal, S. (2006). Approximate Dynamic Programming Methods For An Inventory Allocation Problem Under Uncertainty, *Naval Research Logistics (NRL)*, 53(8), 822-841.

Van Hoesel, C. P. M., Wagelmans, A. P. (2001). Fully Polynomial Approximation Schemes For Single-Item Capacitated Economic Lot-Sizing Problems, *Mathematics of Operations Research*, 26(2), 339-357.

Yu, H., Bertsekas, D. P. (2013). Q-Learning And Policy Iteration Algorithms For Stochastic Shortest Path Problems, *Annals of Operations Research*, 208(1), 95-132.