マルコフ決定過程における近似DPアルゴリズム

中出　康一

マルコフ決定過程は，不確実性をもつシステムの最適制御問題に広く適用可能である．一方で，最適政策を求める際，次元の呪いと呼ばれる問題が指摘されてきた．近年，準最適政策を求める近似 DP アルゴリズムの研究が発展している．本稿では，離散時間マルコフ決定過程で定式化可能な長時間平均費用最小化問題に対する近似 DP アルゴリズム，特にシミュレーションベースの修正政策反復法の概要について述べるとともに，その適用例について触れる．

キーワード：マルコフ決定過程，近似アルゴリズム，修正政策反復法，シミュレーション

1. はじめに

世の中には不確定な要素をもつ事象は数多くある．生産から販売までの製造・物流工程をみても，加工機械の機械故障，工具の不具合による取り替え，顧客の需要，製品を運送するのに必要な時間などさまざまである．これらの変動を考慮しながら，そのときどきの状況に応じてシステムを効率的に運用することが必要がある．

不確定要素をもつシステムの動的最適制御政策を求める際適用される基本的な OR 手法の 1 つがマルコフ決定過程によるモデル化である [1]．マルコフ決定過程に関する研究としては，定式化されたシステムについて，状態に関する値関数の性質を利用して最適政策がもつ性質を導く研究と，政策反復法，値反復法といった最適化アルゴリズムに関する研究などがある．本稿では後者に焦点を当てる．

状態に応じて決定を定める政策を求める方法として広く知られているのは政策反復法である．この方法は，ある弱い条件のもと，有限回で最適政策，すなわち状態に応じたとるべき最適決定を求めることができる．実際，小さいモデルであればほとんどの場合短時間で最適政策を求めることができる．しかし，システムがとりうる状態は，問題が少し複雑になると非常に大きな個数となる．例えば，需要が環境に依存する 3 工程からなる生産システムにおいて，各工程が抱える在庫の数，完成在庫品数，需要環境に関する状態と 5 次元の要素からなり，仮におのおの 20 個の可能性があるとすると，20 の 5 乗，すなわち 3,200,000 状態となる．政策反復法では，反復の際状態数次元の連立一次方程式を解く必要があるため，最適政策を求めることは困難となる．このことは，いわゆる次元の呪いといわれ，以前は大規模な問題には適用できないとされてきた．

値反復法は，相対費用などの値関数を繰り返し計算により最適政策を求める方法であるが，最適政策に収束するまでには大変時間がかかる．とはいえ，反復により値関数自体はある程度更新される．このことを利用し，政策反復法と値反復法を組み合わせた修正政策反復法が提案されている．しかし，それでも数十万〜数千万個の状態のもとでは，厳密な意味での最適政策を求めることはきわめて困難である．

そこで，近年，近似 DP と呼ばれる準最適政策を求める手法に関する研究がなされている [2, 3, 4, 5]．その 1 つとしてここ十数年研究されているニューロ DP は，強化学習の考え方を適用したものである．しかし，一方で，実際に生産システムなどの問題に適用した際，あまり良い結果が得られないことがわかっている．

本稿では，離散時間マルコフ決定過程で定式化可能な，長時間平均費用最小化問題に対するシミュレーションベースの修正政策反復法アルゴリズムについて述べる．さらに，多段階生産システムなどへの応用について示す．

なお，本稿は中出の単著となっているが，愛知工業大学の大野勝久客員教授が中心となり得られた研究成果をもとに，著者が個人の責任として解説を加えたものであることを先に申し上げておく．また，大野 [6] も併せて参考にしていただきたい．

2. マルコフ決定過程と次元の呪い

まずマルコフ決定過程と次元の呪いについて述べる．各時点におけるシステムを表現する状態を s とし，

なかで　こういち
名古屋工業大学大学院社会工学専攻
〒 466–8555 名古屋市昭和区御器所町

状態集合を S とする．状態 s を観測したとき，取りうる決定の集合を $A(s)(s \in S)$ とする．状態 s において決定 $a \in A(s)$ を取るときの 1 期間の期待費用を $r(s,a)$，次期の状態が s' となる確率を $p(s,s',a)$ とする．問題は，無限期間平均期待費用

$$\lim_{t\to\infty}\frac{1}{t}E_\pi\left[\sum_{n=0}^{t-1}c(x_t,a_t)\right]$$

を最小にするような決定政策を求めることである．ここで E_π は履歴に基づいて決定を行う政策 π のもとでの期待値である．さらにシステムが単連結 (simply connected) であると仮定する．ここで単連結とは，各状態について，すべての定常政策で一時的 (transient) であるか，あるいはある定常政策のもとで単一連鎖 (single chain) の状態であるかのいずれかであることを表す．このとき，時間平均費用は初期状態に依存しない定数となり，定常な最適政策が存在する．また，次の最適性方程式の右辺の値を最小化する決定 $a_s \in A(s)$ が状態 s に対する最適決定となり，それらの組 $\{a_s, s \in S\}$ が定常な最適政策となる．

$$g+h(s)=\min_{a\in A(s)}\left\{r(s,\ a)+\sum_{s'\in S}p(s,\ s',\ a)h(s')\right\}.$$

ここで g は平均費用，$h(s)$ はある状態 s_r に対し $h(s_r)=0$ としたときの相対費用であり，初期状態を s としたときの総期待費用に関する状態 s_r との差を表している．

この問題に対する政策反復法とは次のとおりである．ここでは各定常政策で単一連鎖となる場合を示している．定常政策下で複数連鎖 (multi-chain) を形成する可能性があるときは，計算手続きはより複雑になる．

[政策反復法 (PIM)]

1. 初期定常政策 $f=\{f(s), s\in S\}$ を定める．
2.（値決定ルーチン）ある状態 s_r について $h(s_r)=0$ としたときの g と $h(s)$ に関する次の連立一次方程式の解を求める．

$$g+h(s)=r(s,\ f(s))+\sum_{s'\in S}p(s,\ s',f(s))h(s'),\ s\in S.$$

3.（政策改良ルーチン）値決定ルーチンで求めた g と $h(s)$ をもとに，次の式を最小にする決定 a を $f'(s)$ とする．

$$w(s)=\min_{a\in A(s)}\left\{r(s,a)+\sum_{s'\in S}p(s,s',a)h(s')\right\}.$$

すべての $s\in S$ について $f'(s)=f(s)$ ならば $f(s)$ を最適政策として出力．そうでなければ $f'(s)$ を $f(s)$ として 2. に戻る．

したがって，反復ごとに政策を更新するとともに，相対費用 $h(s)$ を更新しながら，最適政策に近づく方法である．この $h(s)$ を求める際，連立一次方程式を解く必要があり，状態数が大きくなると計算時間が膨大になり現実には解けなくなるいわゆる次元の呪い (curse of dimensionality) を引き起こす．

次の修正政策反復法 MPIM では，複数回の逐次計算により $h(s)$ を計算する方法である．

[修正政策反復法 (MPIM)] [7]

1.（初期設定）
ある基準となる状態 $s_r\in S$ に対して $h^0(s_r)=0$ を満たす初期相対費用 h^0 と非負整数 m，初期政策 f^0，正数 ε を定め，$k=0$ とおく．

2.（政策改良ルーチン）
各 $s\in S$ に対して，

$$g^{k+1}(s)=\min_{a\in A(s)}\left\{r(s,a)+\sum_{s'\in S}p(s,s',a)h^k(s')-h^k(s)\right\}$$

を計算する．$f^k(s)$ が $g^{k+1}(s)$ を与えれば，$f^{k+1}(s)=f^k(s)$ とおく．さもなければ，$g^{k+1}(s)$ を与える任意の決定を $f^{k+1}(s)$ とする．

3.（値近似ルーチン）

$$w^0(s)=h^k(s)+g^{k+1}(s),\quad s\in S$$

とおき，$l=0,1,\ldots,m-1$ に対して順次，

$$w^{l+1}(s)=r(s,f^{k+1}(s))+\sum_{s'\in S}p(s,s',f^{k+1}(s))w^l(s')$$

を計算し，$h^{k+1}(s)=w^m(s)-w^m(s_r)$ とおく．すべての $s\in S$ に対して，$|h^{k+1}(s)-h^k(s)|<\varepsilon$ であれば終了．さもなければ，$k=k+1$ として 2. に戻る．

状態ごとに，すでに決定した政策に関する再帰計算を行うことにより $h(s)$ を求めるため，状態数次元の連立一次方程式を解くことに比べれば，計算時間は少なくなる．実際，状態と決定を定めたとき，次期に到達

する可能性のある状態（$p(s,s',f(s))>0$ となる状態 s' の個数）は全状態のうちごくわずかにすぎないことが多いため，この方法は有用である．

しかしながら，先に述べたように，生産工程数が複数になるなどにより簡単に状態数は数百万，数千万に達してしまうため，計算量はやはり大きくなる．また，決定にしても，すべての取りうる決定を毎回考慮して計算すると，政策改良ルーチンにおける計算量はやはり膨大になる．また，すべての状態について $h(s)$ などを計算し，すべての状態と決定の組について推移確率や費用を計算しておくことは空間記憶量を非常に大きくする．

3. 最適政策の構造

マルコフ決定過程の最適政策のもとでは，多くの場合つぎのような性質をもつ（図 1）．

状態集合 S の部分集合 S' の範囲内でしか状態は移動しない．$S^- = S - S'$ に属する状態 s'' から開始しても，いつかは集合 S' に属する状態になり，その後状態 s'' には戻らない（すなわち，s'' は最適政策のもとで一時的となる）．

例えば，在庫・受注残費用を最小化するように，現時点の状態（在庫量，需要の状態など）をもとに最適な発注量を定める問題では，必要以上の在庫を持っていても常に在庫を抱えるだけでメリットがないため，最適政策のもとでもつ最大在庫量は一定以下しかとらない．また，需要が多いことが予想される場合はその前に在庫を確保し，少ない場合は在庫を少なくするなど，在庫量以外の情報に応じて必要な在庫数が変わる．このため，需要の状況をシステムの状態に取り入れた場合，取りうる状態の集合を仮に可能性がある限り大きくとったとしても，最適政策下ではそのうちの一部の状態にしか到達しなくなる可能性が高い．

一方で，政策反復法，修正政策反復法ともにすべての取りうる状態について $h(s)$ を求めようとしている．また，実際の最適政策のもとでは将来的に再度訪問することのない状態についても最適決定を求めようとしている．$h(s)$ の計算，ならびに決定の更新に関する計算を，全状態ではなく，平均費用を小さくする政策のもとで訪問することの多い状態を中心にして，$h(s)$ を更新すれば，より少ない計算で最適に近い政策を導けると考えられる．

また，決定についても，いきなりすべての決定について計算を行い比較すると非常に大きな計算量となる．実際には，決定空間の中には，明らかに最適ではないといえる決定が多く含まれていることがよく起こる．例えば，在庫が多いのに多くの製品を生産することが最適になるとは考えにくい．

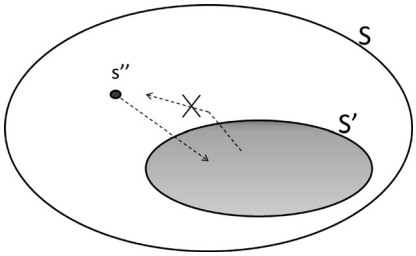

図 1 状態空間と遷移

そこで，システムの構造から見て，ある程度適切であると思われる政策のもとでシミュレーションを行い，訪問した状態とその周辺の状態に関する $h(s)$ と平均費用を推定するとともに，政策改良の際も，すべての決定について計算して比較するのではなく，現在の決定とその決定に近い（近傍の）決定に関してのみ比較を行うことにより効率的に良い決定が生まれてくると考えられる．

この考え方から，シミュレーションベースをもとにした近似最適決定政策を求めるアルゴリズムとして SBMPIM（シミュレーションベース修正政策反復法，Simulation-Based Modified Policy Iteration Method）が考えられてきている．

4. SBMPIM

以下 SBMPIM について述べる．ある状態 s_0 から出発し，システムの状態変化と費用をシミュレートし，訪問した状態 s に対してだけ相対費用 $h(s)$ と平均費用 g を推定する．そのために 3 つの部分状態空間 S_T，S_V，S_U を以下のように準備する．

S_T：1 回の m ステップシミュレーション中に訪問した状態の集合．

S_V：それまで実行したシミュレーションのなかで実際に訪問したことのある状態の集合．

S_U：S_V に属する s に対し $h(s)$ を計算するために新たに必要となる補助的な状態の集合．

[SBMPIM]（図 2，[6] 他）

1.（初期設定）

初期状態 s_0 と望ましい状態 s^* を定め，シミュレーション回数 m，正定数 M，非負数 λ, μ（$\lambda, \mu \leq 1$）および停止基準の正整数 Q と $\varepsilon, \varepsilon' > 0$ を定め，訪問した状態の集合 $S_V = \phi$（空集合），$S_T = \phi$，$S_U = \phi$，累積費用 $TC = 0$，$s = s_0$，$k = l = 1$，

$q=0$ とおき，$f(s_0)$ を状態 s^* へ向かう実行可能な決定と定める．

2.（Schweitzer 変換）
次式を満たす正数 τ

$$0<\tau<\min_{\substack{s\in S,a\in A(s),\\ p(s,s,a)<1}}\{1/(1-p(s,s,a))\}$$

を定め，直接費用 $r(s,a)$，推移確率 $p(s,s',a)$ を

$$r(s,a)\leftarrow \tau r(s,a),$$

$$p(s,s',a)\leftarrow \tau p(s,s',a)+(1-\tau)\delta_{s,s'}$$

と変換する．ここで $\delta_{s,s'}=1,\ s=s';=0,\ s\neq s'$ である．この変換により状態の周期性を防ぐ．

3.（シミュレーション）
3-1: 状態 s で決定 $f(s)$ をとったときの状態推移をシミュレーションし，次期の状態 s' を定め，

$$TC=TC+r(s,f(s)),\quad s=s'$$

と更新する．

3-2: $s\notin S_V$ かつ $s\notin S_U$ ならば，$S_v=S_v\cup\{s\}$，$S_T=S_T\cup\{s\}$，s の訪問回数 $v(s)=1$ とおき，$f(s)$ を状態 s^* へ向かう実行可能な決定と定め，$w(s)=r(s,f(s))$ とおく．

3-3: $s\notin S_V$ かつ $s\in S_U$ ならば，$S_v=S_v\cup\{s\}$，$S_U=S_U-\{s\}$，$S_T=S_T\cup\{s\}$，s の訪問回数 $v(s)=1$ とおく．

3-4: $s\in S_V$ ならば，

$s\notin S_T$ のとき，$S_T=S_T\cup\{s\}$，$v(s)=1$ とおき，$s\in S_T$ ならば，$v(s)=v(s)+1$ と更新する．

3-5: $l=m$ ならばステップ 4 へ．さもなければ $l=l+1$ としてステップ 3-1. へ．

4.（g の推定）
S_T のなかで $\nu(s)$ が最大の s を s_r とおき，1 期あたりの平均費用 $g(k)$ および g を

$$g(k)=TC/m,\quad g=(qg+g(k))/(q+1)$$

により計算する．

5.（$h(s)$ の推定）

$$h(s_r)=(1-\lambda^k v(s_r)/m)w(s_r)+(\lambda^k v(s_r)/m)r(s_r,f(s_r))-g$$

を計算する．

5-1: $s(\neq s_r)\in S_T$ に対して

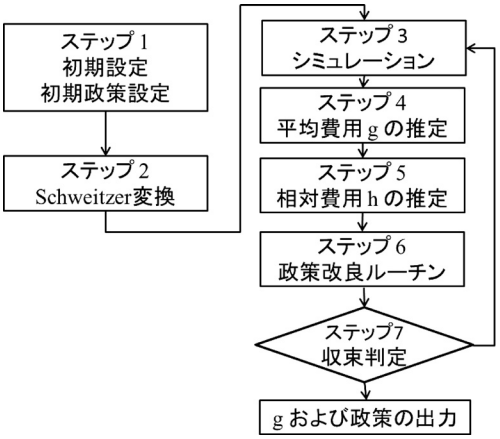

図 2　SBMPIM

$$h(s)=(1-\lambda^k v(s)/m)w(s)+(\lambda^k v(s)/m)r(s,f(s))-g-h(s_r)$$

とおく．

5-2: $s\in(S_V\cup S_U)-S_T$ に対して

$$h(s)=w(s)-g-h(s_r)$$

とおく．

5-3: $h(s_r)=0$ とおく．

6.（政策改良ルーチン）
6-1: $s\in S_V$ に対して

$$w(s)=\min_{a\in N(s,f(s))}\left\{r(s,a)+\sum_{s'\in S}p(s,s',a)h(s')\right\}$$

を計算する．ここで $N(s,f(s))$ は $A(s)$ における $f(s)$ の近傍であり，$p(s,s',a)>0$ となる $s'\notin S_v\cup S_U$ に対しては，$S_U=S_U\cup\{s'\}$ とおき，$f(s')$ を s^* へ向かう実行可能な決定と定め，$w(s')=r(s',f(s'))$，$h(s')=M$ を用いて $w(s)$ を計算する．$f(s)$ が $w(s)$ を与えなければ，$w(s)$ を与える任意の決定として $f(s)$ を改良する．

6-2: $s\in S_U$ に対して

$$w(s)=\min_{a\in N(s,f(s))}\left\{r(s,a)+\sum_{s'\in S}p(s,s',a)h(s')\right\}$$

を計算する．ここで $p(s,s',a)>0$ となる $s'\notin S_V\cup S_U$ に対しては，$h(s')=M$ として $w(s)$ を

計算する．$f(s)$ が $w(s)$ を与えなければ，$w(s)$ を与える任意の決定として $f(s)$ を改良する．

7.（収束判定）

7-1: $|g(k)-g(k-1)| < \varepsilon'$，$k \geq 2$ ならば，$q = q+1$ とおき，ステップ 7-2 へ．さもなければ $q = 0$ とおき，ステップ 7-4 へ．

7-2: $q = 1$ ならば $s_0 = s_r$ とおき，ステップ 7-4 へ．さもなければ，$q < Q$ のとき，ステップ 7-4 へ行き，$q \geq Q$ ならば，ステップ 7-3 へ．

7-3: $\{g(k-q)/\tau, \ldots, g(k)/\tau\}$ の標本分散 S^2 を平均 g/τ を用いて計算し，自由度 q の t 分布の両側 α 点の値を $t_\alpha(q)$ としたとき，

$$t_a(q)S/\sqrt{q+1} < \varepsilon$$

を満たせば停止．最小平均費用 g の 100 $(1-\alpha)$%信頼区間は

$$\left[g/\tau - t_\alpha(q)S/\sqrt{q+1}, g/\tau + t_\alpha(q)S/\sqrt{q+1}\right]$$

であり，準最適政策は $\{f(s), s \in S_V\}$ で与えられる．さもなければステップ 7-4 へ．

7-4: $s = s_0$, $S_T = \phi$，$TC = 0$，$l = 1$，$k = k+1$ とおきステップ 3 へ．

5. 適用例

ここでは，3 つのモデルについて取り上げる．

1. 多段生産・物流システムにおける最適生産・発注政策 [6, 8]

3 工程 JIT 生産・物流システムに対して，単位期間あたりの平均総費用を最小化する最適発注・生産政策を求める問題を考える．各工程は加工前の仕掛品在庫と，加工後の加工済み仕掛品在庫を持つ．後工程から前工程に発注がなされ，工程間には輸送時間がかかる．費用としては，部品および製品の在庫費用および品切れ費用を考えることとする．また，最終工程で最大許容受注残量を超えて失われた需要に対して損失費用がかかるものとする．プル方式として，かんばん方式，基点在庫方式，CONWIP，ハイブリッド方式，拡張かんばん方式を考える．各プル方式のパラメータを最適設定で運用したときの最小平均費用を SBMPIM アルゴリズムによる準最適政策における平均費用と比較し，各プル方式がどれだけ準最適政策に近いかを明らかにしている．数千万の状態数に対し，訪問状態数は数百万である．

2. 刃具保全最適化モデル [9]

2 台の機械に対する刃具保全最適化モデルを考える．

システムは 2 台の直結された機械で構成される．生産方式はロット生産である．機械間にバッファは持たず，機械 1 で加工の終わった部品はすぐに機械 2 で次の加工が開始される．また，いずれかの機械が停止するとシステム全体が停止する．各機械には複数本の刃具がある．複数の部品があり，各部品は機械 1，2 でそれぞれ特定の刃具によって加工される．機械 1 に到着する部品は既約のマルコフ連鎖に従ってロットごとに変化する．

刃具の交換には保全交換と故障による交換の 2 種類がある．保全交換は各ロットの加工開始前にのみ行われる．一方，故障による交換は故障した時点で故障した刃具にのみ行われる．刃具の交換後にはテスト加工が行われる．刃具の状態はその年齢で表される．機械 1 でこれから加工される部品の種類とすべての刃具の状態は各ロットの加工開始前に瞬時に観測され，各刃具を交換するか交換しないかの決定が与える．この問題はセミマルコフ決定過程として定式化され，SBMPIM のセミマルコフ決定過程版を試みている．セミマルコフ決定過程の定式化にしたこととデータ構造を効率的に組んでいなかったため大規模な問題は解けなかったが，SBMPIM により 20,000 状態ほどの問題を解いている．一方で PIM では解けないレベルである．単純な工具ごとの取り替え保全政策と比べ，SBMPIM で求めた政策では一方の工具が故障で取り替える場合はもう一方の工具も同時に保全取り替えをすることで，工具の取り替えの頻度を減らし，ラインの停止時間を減らすことで費用を減少させる効果があることを示している．

3. 需要情報を用いた最適化 [10]

1. のモデルを 2 段階として，需要の情報を用いた場合を扱う．事前に最終工程の需要に関する情報が到着し，その情報を用いて各生産工程は生産を行う．状態として需要の情報が加わることにより，数百万〜1 千万程度の状態数の問題として定式化される．実際シミュレーションで訪問した状態数は全体の 5〜10%程度にとどまる．需要情報がそのまま確定する場合と，後で変化する場合について近似最適政策を求め，情報がないときとの比較を行っている．

6. 改良 SBMPIM と SBMPI

シミュレーションベースの修正政策反復法の枠組みにおいて，効率的なアルゴリズムになるかどうかは，以

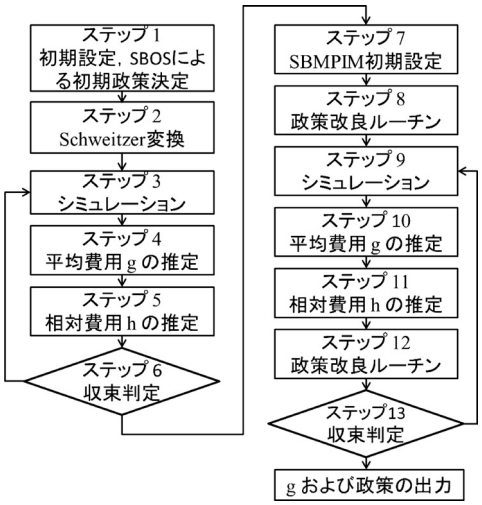

図 3　改良 SBMPIM

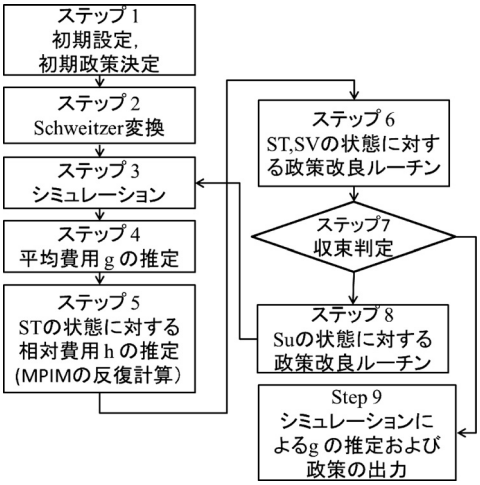

図 4　SBMPI

下の点がかかわってくる．

1. 対象とする問題にとってよい状態，よい政策とはなにかを適切に予測する．

よい政策を初期政策としてとると，$h(s)$ の値もより早く最適政策における値に近づき，収束が早くなると期待できる．

2. $h(s)$ をどのように更新するか．

そこで，前節の多段生産・物流政策について，SBMPIM をさらに効率的にするため以下の方法が編み出された．プル方式のパラメータを最適設定する **SBOS** (simulation-based optimal setting) アルゴリズムをもとに，それにより求めたプル方式を初期政策として SBMPIM を適用する．さらに，その初期政策についてシミュレーションを用いて $h(s)$ や平均費用を前もって計算し，その値をもとに SBMPIM を適用する改良 SBMPIM が提案された（図 3）．実際，[6, 10] などの生産物流システムの最適化では改良 SBMPIM が適用されている．

さらに，SBMPIM のなかの $h(s)$ の更新について，MPIM の値近似ルーチンを組み込んだ SBMPI(Simulation-Based Modified Policy Iteration) も最近提案されている（図 4，[11]）．

7. おわりに

本稿ではマルコフ決定過程における近似 DP アルゴリズムを述べた．これまで示したように，初期政策と $h(s)$ の決定方法はアルゴリズムの性能に大きく影響する．初期政策の決定があまりよくないと $h(s)$ などの計算値が最適政策から離れてしまい，近似最適政策にはなかなか近づかないようである．

$h(s)$ の更新方法もさまざま考えられる．SBMPIM の政策改良ルーチンにおいて，一度でも訪れた状態について政策改良を行っているが，ある程度最適政策に近づいた段階でほとんど訪れることのない状態について政策改良の計算をする必要はないかもしれない．ただ，早い段階で訪問しない状態に対する $h(s)$ の更新を止めてしまうと，$h(s)$ の値が正確さに欠けるため，最適政策には簡単には近づかない可能性がある．また，ここでは議論していないが，政策改良の際の現在の決定に対する近傍の取り方もいろいろな方法が考えらえる．

本稿で述べた枠組みは生産システムだけでなく，多くの分野に応用できると思われる．例えば，弁当などの日持ちしない商品があり，発注は 1 日に 2，3 回行い，納入は約 1 日後といったコンビニ店の最適発注について考える．商品は賞味期限を過ぎると廃棄される．発注済みあるいは店舗に並べた商品について，それぞれ賞味期限日時が異なるため，届いていない商品の各回の注文数やすでに届いてまだ残っている商品数自体を記憶しておく必要があり，仮に単一品の問題でも非常に大きな状態数になってしまうが，このような問題でも近似アルゴリズムが適応可能であると思われる．一般的なモデルに対し，初期政策の決定や $h(s)$ の更新，決定の近傍などに関する計算手続きがまだ完成しているわけではないため，すべてのモデルに簡単に適用できるというところまでは到達していないが，今後のアルゴリズムの発展により，現実的なさまざまな問題にも適用できるものと期待される．

最後に，一言申し上げます．

本稿を再校校正する直前，8 月 8 日に大野勝久先生

が逝去されました．大野先生には，本稿作成にあたり多くの資料をご提供いただきました．入院されて約 2 週間後，今年 4 月末に病院にうかがった際にも USB で資料をいただきましたが，出版を待たずに旅立たれてしまいました．

この場を借りて大野先生に感謝申し上げるとともに，ご冥福をお祈りいたします．

参考文献

[1] M. L. Puterman, *Markov Decision Processes*, John Wiley & Sons, 1994.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific ,1996.

[3] A. Gosavi, *Simulation-based Optimization*: *Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Academic，2003.

[4] W. B. Powell, *Approximate Dynamic Programming*: *Solving the Curses of Dimensionality*, John Wiley & Sons，2007.

[5] X.-R. Cao, *Stochastic Learning and Optimization*, Springer, 2007.

[6] 大野勝久，「サプライチェーンの最適運用：かんばん方式を超えて」，朝倉書店，2011.

[7] K. Ohno and K. Ichiki, “Computing Optimal Policies for Controlled Tandem Queueing Systems,” *Operations Research*, **35**, 121–126, 1987.

[8] K. Ohno, “The Optimal Control of Just-in-time-based Production and Distribution Systems and Performance Comparisons with Optimized Pull Systems,” *European Journal of Operations Research*, **213**, 124–133, 2011.

[9] 吉田昌記，中出康一，大野勝久，ニューロ DP による刃具保全最適化に関する研究，日本経営工学会中部支部研究発表会，13–16, 2011.

[10] 竹村亮祐，中出康一，需要情報をもつ周期観測 2 段生産システムにおける最適発注・生産指示方策，日本オペレーションズ・リサーチ学会中部支部第 40 回支部研究発表会中部品質管理協会，49–52, 2013.

[11] 大野勝久，坊敏隆，田村隆善，近似 DP アルゴリズム SBMPI による生産・物流システムの最適制御，平成 24 年日本オペレーションズ・リサーチ学会秋季研究発表会，ウインクあいち，118–119, 2012.