数理解析研究所講究録 1383 巻 2004 年 64-71

ニューロ DP による多品目在庫管理の最適化

愛知工業大学　大野勝久(Katsuhisa Ohno)
Aichi Institute of Technology,
名古屋工業大学　伊藤崇博(Takahiro Ito)
Nagoya Institute of Technology,
大阪府立大学　石垣智徳(Tomonori Ishigaki)
Osaka Prefecture University,
NTT ドコモ東海　渡辺　誠(Makoto Watanabe)
NTT DoCoMo Tokai

1. 序論

在庫管理問題は，古くは 1913 年の F.W.Harris の経済的発注量(EOQ)の研究に始まり，経営工学分野の中で長年にわたり中心的課題として，多くの研究が行われてきた。今日においては，原材料の調達から消費者にいたる SCM を中心に，多品目在庫管理問題の最適化が切望されている。

本発表ではまず，2 章で離散時間多品目在庫管理問題を平均費用を最小化する時間平均マルコフ決定過程問題として定式化する。動的計画法（以下 DP と略す）の枠組みとマルコフ決定過程（MDP）は，Bellman によって 1950 年代に提案されたが，政策反復法(Policy Iteration Method, PIM)を初めとするアルゴリズムは Howard[1], Puterman[2]等にまとめられている。離散時間多品目在庫管理問題にたいしても，Ohno et al.[3]は，PIM を改善したアルゴリズムを提案している。しかし，MDP は DP 同様，品目数の増加とともに次元の呪いを引き起こし，実用規模の問題を事実上解くことができない。そこで本研究では，人工知能[4]の分野において強化学習(Reinforcement Learning)[5]とも呼ばれている，ニューロ・ダイナミックプログラミング(Neuro- Dynamic Programming，ニューロ DP) [6, 7]の適用を試みる。ニューロ DP は MDP の枠組みの中で，シミュレーション，学習，ニューラルネットワークなどを組み合わせ，大規模な問題に対する近似最適政策を計算する手法として開発されてきた。最近 Das et al.[8]と Gosavi[9]は，MDP を含むより一般的なセミ・マルコフ決定過程（SMDP）の時間平均費用を最小化するアルゴリズムとして SMART, RELAXED-SMART を提案し，保全問題へ適用している。また He et al.[10]は，PIM の値決定ルーチンをシミュレーションで置きかえた SBPI アルゴリズムを提案し，在庫管理問題へ適用している。さらに Gosavi et al.[11]は，SMART に TD(Temporal Differnce) をとり入れた λ−SMART を開発し，航空運賃の収入管理問題へ応用している。

本発表では，まず Ohno et al.[3]に基づき次章で多品目在庫管理問題を説明し，3 章で時間平均マルコフ決定過程（UMDP）への定式化を行い，若干の理論的性質を定理として示す。4 章で修正政策反復法(Modified Policy Iteration Method, MPIM) [12-14]の値決定ルーチンにシミュレーションを用いる SBMPIM アルゴリズム[15]を述べる。5 章では SMART, RELAXED–SMART と SBPI アルゴリズムを簡単に説明する。6 章ではこれらアルゴリズムを 2 品目在庫管理問題へ適用し，SBMPIM, SMART, RELAXED–SMART, SBPI に対する結果をも含めて Ohno et al.[3]による厳密解との比較を行う。

2. 多品目在庫管理問題

発注費用の形が一般的であり，超過需要が部分的もしくはすべてバックログされ，周期的な観測を行う N 品目在庫管理システムを考える。もしロストセールを仮定するならば，品物の納入遅れはないものとする。ここで $\mathbf{x}=(x_1,x_2,\cdots,x_N)^T$ を各期首における発注前の手持ち在庫水準とする。ここで $x_n\in\mathbf{Z}=\{0,\pm1,\pm2,\cdots\}$ ($n\in\mathbf{N}=\{1,2,\cdots,N\}$) であり，$T$ は転置を表す。Johnson[16]を除くすべての論文は連続状態を論じているが，より現実的な離散状態を議論する。発注は $\mathbf{x}$ と発注政策 f によって決定され，$\mathbf{y}=(y_1,y_2,\cdots,y_N)^T\in\mathbf{Z}^N$ まで発注したとき，段取り費用を含む発注費 $K(\mathbf{x},\mathbf{y})$ が発生する。各期の需要 $\mathbf{D}=(D_1,D_2,\cdots,D_N)^T$，$D_n\in\mathbf{Z}_+=\{0,1,2,\cdots\}$ ($n\in\mathbf{N}$) は独立な同一分布 $\phi(\mathbf{y},\mathbf{D})$ に従うものと仮定する。その期に需要 $\mathbf{D}$ が発生したならば，1 期間当たり在庫・品切れ費用 $l(\mathbf{y},\mathbf{D})$ が発生する。したがって，各期における期待在庫・品切れ費用 $L(\mathbf{y})$ は

$$L(\mathbf{y})=\sum_{\mathbf{D}\geq 0}\phi(\mathbf{y},\mathbf{D})\,l(\mathbf{y},\mathbf{D})$$

で与えられる。ここで$\mathbf{x}\leq\mathbf{y}$（$\mathbf{x},\mathbf{y}\in\mathbf{Z}^N$）は$x_n\leq y_n$（$n\in\mathbf{N}$）を表し，$\mathbf{x}<\mathbf{y}$は$\mathbf{x}\leq\mathbf{y}$かつ$\mathbf{x}\neq\mathbf{y}$を意味するものとする。また，次期の期首在庫は$u(\mathbf{y},\mathbf{D})$によって与えられるものとする。例えば，バックログを仮定する場合，$u_B(\mathbf{y},\mathbf{D})=\mathbf{y}-\mathbf{D}$となり，ロストセールを仮定する場合，

$$u_L(\mathbf{y},\mathbf{D})=\left([y_1-D_1]^+,[y_2-D_2]^+,\cdots,[y_N-D_N]^+\right)^T,$$

$$[y_n-D_n]^+=\max\{y_n-D_n,0\}\quad(n\in\mathbf{N})$$

である。

状態$\mathbf{x}$の値からなる状態空間$\mathbf{X}$は倉庫容量$\overline{\mathbf{M}}$と在庫処理環境（例えはバックログ（B），ロストセール（L））によって制約される。例えば，バックログの場合

$$\mathbf{X}_B=\left\{\mathbf{x}\in\mathbf{Z}^N;\sum_{n\in\mathbf{N}}a_n x_n^+\leq\overline{\mathbf{M}}\right\}$$

であり，ロストセールの場合

$$\mathbf{X}_L=\left\{\mathbf{x}\in\mathbf{Z}_+^N;\sum_{n\in\mathbf{N}}a_n x_n\leq\overline{\mathbf{M}}\right\}$$

となる。ここでa_nはn番目の品目1個当たりの大きさであり，$x_n^+=\max\{x_n,0\}$，$n\in\mathbf{N}$である。X_Bは加算集合でありX_Lは有限集合である。以下の条件を仮定する。

条件 1　$\mathbf{x}<\mathbf{y}<\mathbf{z}$であるような任意の在庫水準$\mathbf{x},\mathbf{y},\mathbf{z}\in\mathbf{X}$に対して以下の式が成立する。

$$K(\mathbf{x},\mathbf{z})<K(\mathbf{x},\mathbf{y})+K(\mathbf{y},\mathbf{z})\text{かつ}K(\mathbf{x},\mathbf{x})=0.$$

条件 2　もし$\mathbf{X}$が有限でないならば$\lim_{\mathbf{x}\to-\infty}L(\mathbf{x})=\infty$である。すなわち，総ての実数$M$に対して，$\mathbf{x}\in\mathbf{X}$かつ$\mathbf{x}\geq\mathbf{z}_M$を満たすすべての$\mathbf{x}$に対して$L(\mathbf{x})>M$となる$\mathbf{z}_M\in\mathbf{X}$が存在する。

条件3　もし$\mathbf{X}$が有限でないならば各期の需要は有界である。すなわち，すべての$\mathbf{x}\in\mathbf{X}$に対して

$$\sum_{\mathbf{D}\in\mathbf{B}}\phi(\mathbf{x},\mathbf{D})=1$$

を満たすような有界集合$\mathbf{B}\subset\mathbf{Z}_+^N$が存在する。

条件4　ロストセールの仮定の下での$x_n=0$を除くすべての$\mathbf{x}\in\mathbf{X}$と$n\in\mathbf{N}$に対し$x_n>u_n(\mathbf{x},\mathbf{D})$が確率1で成立する。

Johnson[16]では固定発注費用$K>0$に対して

$$K(\mathbf{x},\mathbf{y})=K\delta(\mathbf{y}-\mathbf{x})+\mathbf{c}\cdot(\mathbf{y}-\mathbf{x})$$

と仮定されている。ここで$\delta(\mathbf{0})=0$であり，$\mathbf{x}>0$ならば$\delta(\mathbf{x})=1$，$\mathbf{c}$はN次元の価格行ベクトルである。明らかにこの発注費用は条件1を満たし，条件2と条件3は可算であるXに対してのみ要求される条件である。また，確率1で需要0をとる品目は在庫管理の対象として無視できるので条件4は一般的な在庫管理システムについて成立する。目的は計画期間が無限にわたる在庫管理システムの単位期間当たりの期待費用を最小にする最適発注政策を見つけることである。

3．　マルコフ決定過程による定式化

前章の多品目在庫管理システムはUMDPに定式化することができ，その最適方程式は次式で与えられる。

$$g(\mathbf{x})+\upsilon(\mathbf{x})=\min_{\mathbf{y}\in Y(\mathbf{x})}\left\{K(\mathbf{x},\mathbf{y})+L(\mathbf{y})+\sum_{\mathbf{D}\in\mathbf{B}}\phi(\mathbf{y},\mathbf{D})\nu(u(\mathbf{y},\mathbf{D}))\right\}\tag{1}$$

$$g(\mathbf{x})=\min_{\mathbf{y}\in\mathbf{Y}}\left\{\sum_{\mathbf{y}\in\mathbf{Y}}\phi(\mathbf{y},\mathbf{D})g(u(\mathbf{y},\mathbf{D}))\right\}\tag{2}$$

ここで

$$\mathbf{Y}=\{\mathbf{y}\in\mathbf{X};\mathbf{y}\geq\mathbf{x}\},$$

$$\mathbf{Y}(\mathbf{x})=\left\{\mathbf{y}\in\mathbf{Y}:g(\mathbf{x})=\sum_{\mathbf{D}\in\mathbf{B}}\phi(\mathbf{y},\mathbf{D})g(u(\mathbf{y},\mathbf{D}))\right\}.$$

である。$g(\mathbf{x})$，$\upsilon(\mathbf{x})$をそれぞれ初期状態が$\mathbf{x}$である時の最小期待平均費用と相対値とおく。有限であるXの場合，確定的な定常マルコフ政策の中に最適政策が存在することがよく知られている。定常政策fは$\mathbf{X}$の部分集合σとσから$\mathbf{X}$への写像$S(\cdot)$を用いて$f=(\sigma,S(\cdot))$と表わすことができる。σは$(\sigma,S(\cdot))$政策は在庫水準$\mathbf{x}$において次のように発注を決定する。

$\mathbf{x}\in\sigma$のとき　　$S(\mathbf{x})$まで発注する。

$\mathbf{x}\in\sigma^c\equiv\mathbf{X}-\sigma$のとき　発注しない。

したがって，σを発注状態集合，σ^cを非発注状態集合と呼ぶことにする。

定理1　最適発注政策$(\sigma,S(\cdot))$は以下を満たす。

i) もし$\mathbf{X}$が有限でないならば

$$\sigma^c\subset\left(\sigma^0\right)^c=\left\{\mathbf{x}\in\mathbf{X};\mathbf{x}\geq\mathbf{z}_{g^0}\right\},\tag{3}$$

である。ここで$\mathbf{z}_{g^0}$は仮定2のMを以下で与えられるg^0によって置き換えたものである。

$$g^0=\min_{\mathbf{y}\in\mathbf{X}}\left\{L(\mathbf{y})+\sum_{\mathbf{D}\in\mathbf{B}}\phi(\mathbf{y},\mathbf{D})K(u(\mathbf{y},\mathbf{D}),\mathbf{y})\right\}.\tag{4}$$

ii)　$S(\mathbf{x})\in\sigma^c$，$\mathbf{x}\in\sigma$.

iii)　$\nu(\mathbf{x})=K(\mathbf{x},S(\mathbf{x}))+\upsilon(S(\mathbf{x}))$，$\mathbf{x}\in\sigma$.

(5)

もし $\mathbf{X}$ が有限でないならば $\mathbf{X}^0$ を

$$\left(\sigma^0\right)^c \cup \left\{ u(\mathbf{x},\mathbf{D}) \quad for\ all\ \mathbf{x} \in \left(\sigma^0\right)^c \text{ かつ } \mathbf{D} \in \mathbf{B} \right\}$$

とおき，$\mathbf{X}^0$ の状態数を N^0 とする。定理 1 は最適政策の候補となるようなすべての発注政策の下で $\mathbf{X}^0$ がすべての再帰的状態を含むことを意味している。逆に $\mathbf{X}^0$ の外側のすべての状態は最適政策の下で常に過渡的状態であり，この状態における発注政策は最小平均期待費用に影響しない。したがって，有限集合 $\mathbf{X}^0$ のみに対して最適政策を探せば十分である。$\mathbf{x} \notin \mathbf{X}^0$ に対する最適政策 $S(\mathbf{x})$ は(5)式より容易に計算できる。次の定理 2 では $\mathbf{X}$ が有限であるとき，$\mathbf{X}^0$ は $\mathbf{X}$ を意味するものとする。

定理 2　最小平均期待費用 $g(\mathbf{x})$ は $\mathbf{x} \in \mathbf{X}$ に対して定数である。すなわち，$g(\mathbf{x}) = g$ であり，最適方程式(1)と(2)は次式に帰着する。

$$g + v(\mathbf{x}) = \min_{\mathbf{y} \in \mathbf{Y}} \left\{ K(\mathbf{x},\mathbf{y}) + L(\mathbf{y}) + \sum_{\mathbf{D} \in \mathbf{B}} \phi(\mathbf{x},\mathbf{D}) v(u(\mathbf{x},\mathbf{D})) \right\}, \quad \mathbf{x} \in \mathbf{X}^0, \quad (6)$$

ここで右辺の最小値を与える決定が最適政策である。

4．修正政策反復法とニューロ DP アルゴリズム

最適性方程式(6)を解くアルゴリズムが PIM, MPIM であり，値反復法（value iteration method），線形計画法（linear programming）である[1,2,12-14]。以下，表記を簡単化するため

$$r(\mathbf{x},\mathbf{y}) = K(\mathbf{x},\mathbf{y}) + L(\mathbf{y})$$
$$\mathbf{x}' = u(\mathbf{y},\mathbf{D})$$

とおく。

[MPIM] [14]

ステップ 1：$v_0(s_r) = 0$ をみたす初期ベクトル v^0，非負整数 m，初期政策 f^0，正数 ε を定め，$k = 0$ とおく。

ステップ 2：(政策改良ルーチン)　各 $\mathbf{x} \in \mathbf{X}^0$ に対して

$$g^{k+1}(\mathbf{x}) = \min_{\mathbf{a} \in K(s)} \left\{ r(\mathbf{x},\mathbf{y}) + \sum_{\mathbf{D} \in \mathbf{B}} \phi(\mathbf{x},\mathbf{D}) v^k(\mathbf{x}') - v^k(\mathbf{x}) \right\}$$

を計算し，$f^k(\mathbf{x})$ が $g^{k+1}(\mathbf{x})$ を与えれば，$f^{k+1}(\mathbf{x}) = f^k(\mathbf{x})$ とおき，さもなければ，$g^{k+1}(\mathbf{x})$ を与える任意の決定を $f^{k+1}(\mathbf{x})$ とする。

ステップ 3：(値近似ルーチン)

$$w^0(\mathbf{x}) = v^k(\mathbf{x}) + v^{k+1}(\mathbf{x}), \quad \mathbf{x} \in \mathbf{X}^0$$

とおき，$l = 0,1,\cdots m-1$ に対して順次，

$$w^{l+1}(\mathbf{x}) = r\left(\mathbf{x}, f^{k+1}(\mathbf{x})\right) + \sum_{\mathbf{D} \in \mathbf{B}} \phi(\mathbf{x},\mathbf{D}) w^l(\mathbf{x}'), \quad \mathbf{x} \in \mathbf{X}^0$$

を計算し，

$$v^{k+1}(\mathbf{x}) = w^m(\mathbf{x}) - w^m(\mathbf{x}_r), \quad \mathbf{x} \in \mathbf{X}^0$$

とおく。すべての $\mathbf{x}$ に対して，$\left|v^{k+1}(\mathbf{x}) - v^k(\mathbf{x})\right| < \varepsilon$ であれば終了。最適政策は $f^{k+1}(\mathbf{x})$，最小平均費用は $g^{k+1}(\mathbf{x}_r)$，相対費用は $v^{k+1}(\mathbf{x})$ で与えられる。さもなければ，$k = k+1$ として，ステップ 2 へ。

MPIM は比較的状態数の大きな MDP に対しても有効であるが，DP 同様，品目数の増加とともに次元の呪いを引き起こし，多品目問題を解くことは困難である。したがって，MPIM において状態空間 $\mathbf{X}^0$ の全ての状態にたいして値近似ルーチンを実行することは実際的ではなく，シミュレーションを用いることが考えられる。すなわち，実際によく生起する初期状態 $\mathbf{x}_0$ 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，訪問した状態の集合 $\mathbf{X}_v$ にたいしてだけ相対費用 $v(\mathbf{x})$ を推定する。このニューロ DP アルゴリズムを SBMPIM （Simulation-Based Modified Policy Iteration Method）と呼ぶことにする[15]。

[SBMPIM] [15]

ステップ 1：初期状態 $\mathbf{x}_0$ と望ましい状態 $\mathbf{x}^*$ を定め，シミュレーション回数 m および λ（$0 \le \lambda \le 1$）を定めて，訪問した状態の集合 $\mathbf{X}_v = \mathbf{X}_T = \phi$（空集合），累積費用 $TC = 0$，$\mathbf{x} = \mathbf{x}_0$，$k = l = 1$ とおく。

ステップ 2：$\mathbf{x} \notin \mathbf{X}_v$ ならば，$\mathbf{X}_v = \mathbf{X}_v \cup \{\mathbf{x}\}$，$\mathbf{X}_T = \mathbf{X}_T \cup \{\mathbf{x}\}$，$\mathbf{x}$ の訪問回数 $v(\mathbf{x}) = 1$ とおき，$f(\mathbf{x})$ を状態 $\mathbf{x}^*$ へ向かう実行可能な決定と定める。$\mathbf{x} \in \mathbf{X}_v$ ならば，$\mathbf{x} \notin \mathbf{X}_T$ のとき，$\mathbf{X}_T = \mathbf{X}_T \cup \{\mathbf{x}\}$，$v(\mathbf{x}) = 1$ とおき，$\mathbf{x} \in \mathbf{X}_T$ ならば，

$$v(\mathbf{x}) = v(\mathbf{x}) + 1$$

と更新する。状態 $\mathbf{x}$ で決定 $f(\mathbf{x})$ 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 $\mathbf{x}'$ を定める。

$$TC = TC + r(\mathbf{x}, f(\mathbf{x}))$$
$$\mathbf{x} = \mathbf{x}'$$

と更新し，$l = m$ ならばステップ 3 へ。さもなければ $l = l+1$ としてステップ 2 へ。

ステップ 3：(g の推定) 平均費用 g を次式により推定する。

$$g = TC/m$$

ステップ 4：($v(\mathbf{x})$ の推定) $\mathbf{X}_v$ のなかで $\mathbf{x}_r$ を定め，

$$v(\mathbf{x}_r) = (1 - \lambda v(\mathbf{x}_r)/m)(w(\mathbf{x}_r) - g) + (\lambda v(\mathbf{x}_r)/m)(r(\mathbf{x}_r, f(\mathbf{x}_r)) - g)$$

を計算し，$\mathbf{x}(\neq \mathbf{x}_r) \in \mathbf{X}_v$ にたいして

$v(\mathbf{x})=(1-\lambda v(\mathbf{x})/m)(w(\mathbf{x})-g)+(\lambda v(\mathbf{x})/m)(r(\mathbf{x},f(\mathbf{x}))-g)-v(\mathbf{x}_r)$ を計算し，$v(\mathbf{x}_r)=0$ とおく。ただし，$k=1$ のときには $v(\mathbf{x}_r)=r(\mathbf{x}_r,f(\mathbf{x}_r))-g$，$v(\mathbf{x})=r(\mathbf{x},f(\mathbf{x}))-g-v(\mathbf{x}_r)$ である。

ステップ 5：（政策改良ルーチン）$\mathbf{x}\in\mathbf{X}_v$ にたいして

$$w(\mathbf{x})=\min_{\mathbf{y}\in N(\mathbf{x},f(\mathbf{x}))}\left\{r(\mathbf{x},\mathbf{y})+\sum_{\mathbf{D}\in\mathbf{B}}\phi(\mathbf{y},\mathbf{D})v(\mathbf{x}')\right\}$$

を計算し，$v(\mathbf{x})=0$ とおく。ここで $N(\mathbf{x},f(\mathbf{x}))$ は $f(\mathbf{x})$ の近傍であり，$\phi(\mathbf{y},\mathbf{D})>0$ となる $\mathbf{x}\notin\mathbf{X}_v$ にたいしては，$\mathbf{X}_v=\mathbf{X}_v\cup\{\mathbf{x}'\}$，$v(\mathbf{x}')=0$ とおき，$f(\mathbf{x}')$ を $\mathbf{x}^*$ へ向かう実行可能な決定と定める。$w(\mathbf{x}')=r(\mathbf{x}',f(\mathbf{x}'))$ とおき，

$$v(\mathbf{x}')=r(\mathbf{x}',f(\mathbf{x}'))-r(\mathbf{x}_r,f(\mathbf{x}_r))$$

として，$w(\mathbf{x})$ を計算する。$f(\mathbf{x})$ が $w(\mathbf{x})$ を与えなければ，$w(\mathbf{x})$ を与える任意の決定として $f(\mathbf{x})$ を改良する。k が停止回数に達すれば終了。さもなければ $\mathbf{X}_T=\phi$，$TC=0$，$l=1$，$k=k+1$ とおきステップ 2 へ。

5．SMART と SBPI アルゴリズム

MPIM と SBMPIM を最適制御問題に適用するに先立ち，既存の NDP アルゴリズムを簡単に紹介する。

[SMART] [8]

ステップ 1：全ての $\mathbf{x}\in\mathbf{X}$ と $\mathbf{y}\in K(\mathbf{x})$ にたいして Q-factor $Q_{new}(\mathbf{x},f)=Q_{old}(\mathbf{x},f)=0$，累積費用 $TC=0$，累積時間 $T=0$，平均費用 $g=0$，反復回数 $k=0$ とおき，学習率（learning rate）と探検確率（exploration probability）をステップ 2 で定める定数 $(\alpha_0,\alpha_\tau,p_0,p_\tau)$ を与える。

ステップ2：反復 k で状態 $\mathbf{x}$ にいれば，学習率 α_k，探検確率 p_k を

$$\alpha_k=\alpha_0(\alpha_\tau+k)/(k^2+k+\alpha_\tau),$$
$$p_k=p_0(p_\tau+k)/(k^2+k+p_\tau)$$

として定める。

ステップ 3：高い確率 $(1-p_k)$ で $Q_{new}(\mathbf{x},\mathbf{y})$ を最小にする決定 $\mathbf{y}^*$ を選択し，確率 p_k で $\mathbf{y}^*$ を除く $K(\mathbf{x})$ からランダムに $\mathbf{y}$ を選択する。

ステップ 4：選択された決定 $\mathbf{y}$ でシミュレーションを行い，状態 $\mathbf{x}'$ へ推移すれば，直接費用 $r(\mathbf{x},\mathbf{x}',\mathbf{y})$ がかかる。

ステップ 5：$Q_{new}(\mathbf{x},\mathbf{y})$ を次式により更新する。

$$Q_{new}(\mathbf{x},\mathbf{y})=(1-\alpha_k)Q_{old}(\mathbf{x},\mathbf{y})+\alpha_k\left\{r(\mathbf{x},\mathbf{x}',\mathbf{y})-g+\min_{\mathbf{y}'\in K(\mathbf{x}')}Q_{old}(\mathbf{x}',\mathbf{y}')\right\}$$

ステップ 6：ステップ 3 で決定 $\mathbf{y}^*$ を選択したならば，TC と g を更新する。

$$TC=TC+r(\mathbf{x},\mathbf{x}',\mathbf{y}^*)$$
$$T=T+1$$
$$g=TC/T$$

ステップ 7：$Q_{old}(\mathbf{x},\mathbf{y})=Q_{new}(\mathbf{x},\mathbf{y})$ と更新する。

ステップ 8：k が停止回数に達すれば終了。さもなければ $k=k+1$，$\mathbf{x}'$ を $\mathbf{x}$ としてステップ 2 へ。

Gosavi[9]は SMART が必ずしも収束しないことを示し，その改良版 RELAXED–SMART を提案している。SMART と RELAXED–SMART は SMDP にたいするアルゴリズムであるが，ここでは MDP にたいするものに修正している。

[RELAXED–SMART] [9]

ステップ 3～5，7，8 は[SMART]と同じである。

ステップ 1：Q-factor $Q_{new}(\mathbf{x},\mathbf{y})=Q_{old}(\mathbf{x},\mathbf{y})=0$，$TC=0$，$T=0$，$g=0$，$k=0$ とおき，パラメータ α_0，p_0，β_0 を与える。

ステップ 2：反復 k で状態 $\mathbf{x}$ にいれば，α_k，p_k，β_k を

$$\alpha_k=a_0/k,\quad p_m=p_0/m,\quad \beta_k=\beta_0/k$$

として定める。

ステップ 6：ステップ 3 で決定 $\mathbf{y}^*$ を選択したならば，TC，T，g を次式で更新する。

$$TC=(1-\beta_k)TC+\beta_k r(\mathbf{x},\mathbf{x}',\mathbf{y}^*)$$
$$T=(1-\beta_k)T+\beta_k$$
$$g=TC/T$$

一方，He et al.[10]は PIM の値決定ルーチンをシミュレーションで置きかえた SBPI（Simulation Based Policy Iteration）アルゴリズムを提案している。

[SBPI アルゴリズム] [10]

ステップ 1：初期政策 $\{f_0(\mathbf{x});\mathbf{x}\in\mathbf{X}\}$ を定め，$k=0$ とおく。

ステップ 2：（値決定ルーチン）

2-a：（g^k の推定）

i) 初期状態 $\mathbf{x}_0$ からシミュレーションにより $\mathbf{x}_1,\cdots,\mathbf{x}_m$ を生成する。

ii) $g^k=0$ とおき，$n=0,\cdots,m-1$ にたいして $(\mathbf{x}_n,\mathbf{x}_{n+1})$ の推移に伴う g^k を次式で更新する。

$$g^k=(1-1/(n+1))g^k+(1/(n+1))r(\mathbf{x}_n,\mathbf{x}_{n+1},f^k(\mathbf{x}_n))$$

2-b：（$v^k(\mathbf{x})$ の推定）

i) ステップ 2-a で行ったシミュレーションにおいて訪問回数が最も多い状態を $\mathbf{x}^*$ とおく。

ii) 過渡状態 $\mathbf{x}_0$ から出発し，状態 $\mathbf{x}^*$ へ至るトラジェクトリーをシミュレーションにより L 本生成する。

iii) 1 本目のトラジェクトリ $\left(\mathbf{x}_0,\mathbf{x}_1,\cdots,\mathbf{x}_N=\mathbf{x}^*\right)$, $l=1,\cdots,L$, にたいして，推移 $(\mathbf{x}_n,\mathbf{x}_{n+1})$ に伴う $w(\mathbf{x}_i)$, $i=1,\cdots,n$, を次式により更新する。

$$w(\mathbf{x}_i)=w(\mathbf{x}_i)+\gamma_i\lambda^{n-i}d_n$$

ここで，γ_i はそのトラジェクトリ中で s_i を訪問した回数の逆数であり，$0\le\lambda\le1$，$d_n=r\left(\mathbf{x}_n,\mathbf{x}_{n+1},f^k(\mathbf{x}_n)\right)-g^k+w(\mathbf{x}_{n+1})-w(\mathbf{x}_n)$ である。

iv) $v^k(\mathbf{x})=w(\mathbf{x})-w(\mathbf{x}_r)$, $\mathbf{x}\in\mathbf{X}$

ステップ 3：(政策改良ルーチン)

$$f^{k+1}(\mathbf{x})=\arg\min_{\mathbf{y}\in K(\mathbf{x})}\left\{r(\mathbf{x},\mathbf{y})+\sum_{\mathbf{D}\in\mathbf{B}}\phi(\mathbf{y},\mathbf{D})v^k(\mathbf{x}')\right\},$$

$\mathbf{x}\in\mathbf{X}$

ステップ 4：$f^{k+1}(\mathbf{x})=f^k(\mathbf{x})$, $\mathbf{x}\in\mathbf{X}$ ならば停止。最適政策は $f^k(\mathbf{x})$ である。さもなければ $k=k+1$ としてステップ 2 へ。

6. 数値実験

各アルゴリズムは VB（Visual Basic）によりプログラムし，DOS/V 機（CPU : Pentium4 2.4GHz, メモリ : 2GB）を用いて計算を行った。

6．1 数値例

・各品目の容積はすべて 1

・品目数 ； 2

・在庫容量； 2 3

・一期あたりの i 番目の需要 D_i ($i=1,2$) の確率 $\phi_i(D_i)$ ；

$\phi_1(0)=1/27$, $\phi_1(1)=2/9$, $\phi_1(2)=4/9$, $\phi_1(3)=8/27$

$\phi_2(0)=1/16$, $\phi_2(1)=1/4$, $\phi_2(2)=3/8$, $\phi_2(3)=1/4$, $\phi_2(4)=1/16$

・発注費用；

$$K(\mathbf{x},\mathbf{y})=15\delta(\mathbf{y}-\mathbf{x})+10\delta(y_1-x_1)+5\delta(y_2-x_2)+6(y_1-x_1)+3(y_2-x_2)$$

・保管・品切れ費用；

$$l(\mathbf{y},\mathbf{D})=2[y_1-D_1]^+ +2[y_2-D_2]^+ +21[D_1-y_1]^+ +14[D_2-y_2]^+$$

6．2 SBMPIM と従来のアルゴリズムとの比較

SBMPIM および既存のアルゴリズムのパラメータを表 1 に示す。表 1 のパラメータのもとで各アルゴリズムにより計算したときの g および計算時間を表 2 に示す。

表 3 は MPIM による最適政策である。また，訪問回数の多い状態における MPIM および各アルゴリズムの政策との比較を表 4 に示す。表中「一致」の列には MPIM と政策が等しい場合に○，そうでない場合に×を記入している。また，「政策一致数」の行は全 300 状態中で MPIM と政策が一致した状態数を示している。

表 1 アルゴリズムに使用するパラメーター覧

アルゴリズム	パラメータ
SBMPIM	m=10000, $\lambda=0.1$, $x_0=(11,12)$, $x^*=(8,9)$, 停止回数=50
SMART	m=1000000, $\alpha_0=0.9$, $p_0=0.3$, $\alpha_\tau=10000$, $p_\tau=10000$
RELAXED-SMART	m=1000000, $\alpha_0=0.9$, $p_0=0.3$
SBPI	m=10000, L=1, $\lambda=0.9$

表 2 各アルゴリズムの g および計算時間

アルゴリズム	g	計算時間(秒)
MPIM	40.907	0.35
SBMPIM	40.910±0.043	1.11
SMART	62.626	3.98
RELAXED-SMART	65.500	4.01
SBPI	43.217	1.92

3 品目在庫管理問題に対して MPIM と SBMPIM により計算を行った結果を表 5 に示す。ここで，2 品目の場合と異なる SBMPIM のパラメータは $x_0=(7,8,8)$, $x^*=(5,6,6)$ である。また，需要およびコストのパラメータは品目 1，2 に関しては 2 品目の場合と同じであり，品目 3 に関しては，需要パラメータは品目 1 と同じ，コストパラメータは品目 2 と同じである。多品目での SBMPIM の優位性が示されている。

表 5 3 品目問題の g および計算時間

アルゴリズム	g	計算時間(秒)
MPIM	54.890	45.37
SBMPIM	54.904±0.041	113.27

謝辞

本研究の一部は，日本学術振興会平成１４－１５年度科学研究費補助金（基盤研究（B）（1），課題番号１４３８０１８５）および（基盤研究（C）（2），課題番号１４５８０４７７）の助成を受けてなされたものである。

参考文献

[1] R. A. Howard: *Dynamic Programming and Markov Processes*, MIT Press (1960)（関根，羽島，森共訳「ダイナミックプログラミングとマルコフ過程」，培風館，1971）

[2] M. L. Puterman: *Markov Decision Processes*, John Wiley& Sons (1994)

[3] K. Ohno, T. Ishigaki and T. Yoshii: "*A New Algorithm for a Multi-item Periodic Review Inventory System*", ZOR-Math. Methods of Oper. Res. Vol. 39, pp. 349-364, 1994.

[4]S. Russell and P. Norvig, 古川康一 訳「エージェント アプローチ 人工知能」, 共立出版, 1997.

[5] R. S. Sutton and A. G. Barto: *Reinforcement Learning*, MIT Press (1998)（三上，皆川共訳「強化学習」，森北出版，2000）

[6] D. P. Bertsekas and J. N. Tsitsiklis: *Neuro-Dynamic Programming*, Athena Scientific (1996)

[7] R. V. Roy: "Neuro-dynamic programming: overview and recent trends," pp.431-459, in E. A. Feinberg and A. Schwartz ed. Handbook of Markov Decision Processes, Kluwer Academic Publishers (2002)

[8] T. K. Das, A. Gosavi, S. Mahadevan and Nich. Marchalleck: "Solving semi-Markov decision problem using average reward reinforcement learning", *Management Science*, Vol. 45, No.4, pp560-574 (1999)

[9] A. Gosavi: Doctor Thesis, http://faculty.uscolo.edu/gosavi/thesis.html (1999)

[10] Y. He, M. C. Fu and S. I. Marcus: "A Simulation- based policy iteration algorithm for average cost unichain Markov decision processes", M. Laguna and J. L. G. Velarde ed, *Computing Tools for Modeling, Optimization and Simulation*, Kluwer Academic, pp.161-182 (2000)

[11] A. Gosavi, N. Bandla and T. K. Das: "A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking," *IIE Transactions*, Vol. 34, pp.729-742 (2002)

[12] 大野勝久："マルコフ決定過程"，システムと制御，Vol. 29，No. 6，pp.333-341 (1985)

[13] K. Ohno and K. Ichiki: "Computing optimal policies for controlled tandem queueing systems", *Operations Research*, Vol. 35, No. 1, pp.121-126 (1987)

[14] K. Ohno: "Modified policy iteration algorithm with nonoptimality tests for undiscounted Markov decision process", Working Paper, Dept. of Information System and Management Science, Konan University, Japan (1985)

[15] 大野勝久，八嶋憲司，伊藤崇博："ニューロ・ダイナミックプログラミングによる生産ラインの最適制御に関する研究"，日本経営工学会論文誌，Vol. 54，No. 5，pp. 316-325 (2003)

[16] E. L. Johnson: "Optimality and Computation of (σ,S) Policies in the Multi-item Infinite Horizon Inventory Problem", *Management Science*, Vol. 13, pp. 475-491 (1967)

[17] 大野勝久，田村隆善，森健一，中島健一：「生産管理システム」，朝倉書店（2002）

[18] 石塚陽，山下英明：「サンプルパス最適化の確率的離散事象システムへの適用」，オペレーションズ・リサーチ，Vol. 46，No. 4，pp.195-201 (2001)

表3　最適政策

品目2																								
23	−																							
22	−	−																						
21	35	**−35**																						
20	34	−	**−34**																					
19	32	−	−	**−32**																				
18	30	−	−	−	**−30**																			
17	28	28	−	−	−	**−28**																		
16	26	26	−	−	−	−	**−26**																	
15	24	24	−	−	−	−	−	**−24**																
14	22	22	−	−	−	−	−	**−22**	−															
13	20	20	−	−	−	−	−	**−20**	−	−														
12	18	18	−	−	−	−	−	**−18**	−	−	−													
11	16	16	−	−	−	−	−	**−16**	−	−	−	−												
10	14	14	−	−	−	−	−	**−14**	−	−	−	−	−											
9	12	12	−	−	−	−	−	**−12**	−	−	−	−	−	−										
8	10	10	−	−	−	−	**−10**	−	−	−	−	−	−	−	−									
7	8	8	−	−	−	−	−	**8**	**9**	**11**	**13**	**15**	**17**	**19**	**21**	**23**	**25**							
6	1	1	−	−	−	−	**1**	**7**	−	−	−	−	−	−	−	−	−	**−27**						
5	6	6	−	−	**3**	**5**	**6**	−	−	−	−	−	−	−	−	−	−	−	**−29**					
4	4	−	−	**2**	−	**4**	−	−	−	−	−	−	−	−	−	−	−	−	−	**−31**				
3	1	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	**−33**			
2	1	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−		
1	1	1	−	−	−	5	1	7	9	−	−	−	−	−	−	−	−	−	−	−	−	−	−	−
0	1	1	1	2	3	5	1	7	9	11	13	15	17	19	21	23	25	27	29	31	33	−	−	−
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	品目1

表2は最適政策を示す。表中の数字と，同表中のゴシック体の数字が対応している。
例えば，状態　[7, 1]の数字“7”は“**7**”に対応し，つまり状態 [7, 6]まで発注するということを示す。
また，“−”は発注しないことを示す。

表4　各アルゴリズムの政策の比較

状態		MPIM		SBMPIM			SBPI		
品目0	品目1	発注0	発注1	発注0	発注1	一致	発注0	発注1	一致
0	0	6	6	6	6	○	5	6	×
4	4	0	0	0	0	○	0	0	○
0	1	6	5	6	5	○	5	5	×
1	0	5	6	5	6	○	4	6	×
3	4	0	0	0	0	○	0	0	○
4	3	0	0	0	0	○	0	0	○
3	3	0	0	0	0	○	0	0	○
2	2	0	0	0	0	○	0	3	×
1	1	5	5	5	5	○	4	5	×
4	5	0	0	0	0	○	0	0	○
1	2	0	0	0	0	○	4	4	×
0	2	6	4	6	4	○	5	4	×
2	0	4	6	4	6	○	0	5	×
2	1	0	0	0	0	○	0	4	×
3	2	0	0	0	0	○	0	0	○
政策一致数（300状態中）				300			136		

状態		MPIM		SMART			RELAXED-SMART		
品目0	品目1	発注0	発注1	発注0	発注1	一致	発注0	発注1	一致
0	0	6	6	2	1	×	2	1	×
4	4	0	0	0	0	○	0	0	○
0	1	6	5	1	1	×	1	0	×
1	0	5	6	0	0	×	0	1	×
3	4	0	0	0	0	○	0	0	○
4	3	0	0	0	0	○	0	0	○
3	3	0	0	0	0	○	0	0	○
2	2	0	0	0	0	○	0	0	○
1	1	5	5	0	0	×	0	0	×
4	5	0	0	0	0	○	0	0	○
1	2	0	0	0	0	○	0	0	○
0	2	6	4	0	0	×	1	0	×
2	0	4	6	0	0	×	0	0	×
2	1	0	0	0	0	○	0	0	○
3	2	0	0	0	0	○	0	0	○
政策一致数（300状態中）				242			237		