数理解析研究所講究録
第1559巻 2007年 139-155

ニューロ・ダイナミックプログラミングとその応用

愛知工業大学・経営情報科学部　大野勝久(Katsuhisa Ohno)
Faculty of Management and Information Science,
Aichi Institute of Technology

1. 序論

動的計画法（Dynamic Programming，以下DPと略す）の枠組みとマルコフ決定過程(Markov Decision Process, MDP)は，周知のように，Bellmanによって1950年代に提案され，政策反復法(Policy Iteration Method，PIM)，値反復法(Value Iteration Method，VIM)を初めとするアルゴリズムがHoward[1]，Puterman[2]等にまとめられている。そして，修正政策反復法(Modified Policy Iteration Method，MPIM)が，比較的状態数の大きな生産ラインの最適制御問題に有効であることが示されている[3-5]。また，離散時間多品目在庫管理問題にたいしても，PIMを改善した効率的なアルゴリズムが提案されている[6]。しかし，MDPはDP同様，工程数あるいは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ものとされていた。

近年，人工知能[7]の分野において，強化学習(Reinforcement Learning)[8,9]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）[10,11]が不確実な環境下における多方面の最適制御問題へ適用されている。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MDP問題に対する近似最適政策を計算する手法であり，そのアルゴリズムが国内外で活発に開発されている。

Das et al.[12]とGosavi[13]は，MDPを含むより一般的なセミ・マルコフ決定過程（Semi-Markov Decision Process, 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.[14]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.[15]は，SMARTに時間的差分（Temporal Difference, TD）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。なお，この分野の専門書として，最近Das[16]およびChang et al.[17]が出版されている。

しかし，これら既存のアルゴリズムを最も簡単な単一品種単一工程生産ラインの最適制御問題へ適用した結果，かんばん方式にも全く歯が立たない制御政策しか生成できなかった。そこで，著者ら[18]はニューロDPアルゴリズムSBMPIMを開発し，同じ生産ラインへ適用し，最適制御政策が得られることを確認した。そして，2工程生産ラインへ適用した結果，最適かんばん枚数のもとでのかんばん方式より8%以上コストを低減できた。さらに，SBMPIMアルゴリズムを3工程以上の生産・物流システムへ適用できるように改良し，シミュレーションによりプル方式を最適に設定した性能と比較して，各プル方式がSBMPIMにより計算された準最適政策にどれだけ近いかを調べている[19,20]。ここで対象としたプル方式

は，かんばん方式[21]，基点在庫方式[22,23]，CONWIP[24]，ハイブリッド方式[25]である。

本論文では，まず次章で Sutton and Barto[8]に基づき強化学習と時間平均マルコフ決定過程（Undiscounted Markov Decision Process，UMDP）を説明し，3章で既存のニューロ DP アルゴリズムとして SMART[12,13]と SBPI[14]を紹介する。4 章で MPIM に基づいたニューロ DP アルゴリズムとして SBMPIM の改良版[19,20]を紹介し，5 章で単一品種単一工程生産ラインの最適制御問題を対象に，これらニューロ DP アルゴリズムの比較を行う。6 章では，[19,20]では考慮していなかった一般化かんばん方式[26,27,28]をも取り入れたプル方式間の比較を，SBMPIM による準最適政策を基準にして行う。

2. 強化学習と UMDP

強化学習は，[8]によれば，「エージェントの利得を最大にするためには，どのようにして状況に基づく動作選択を行うか？」を環境との相互作用から学習する。この点が，メタヒューリスティクスと異なる。そして，長期的な目標を達成するために，環境との相互作用から学習を行うときに発生する計算上の問題を取り扱う。その構成要素は

1. 環境モデル（environment model），状態
2. 学習主体であるエージェント（Learning agent）
3. エージェントの決定（action），政策(policy)
4. 利得関数（reward function），環境からの反応（environmental response）
5. 最大化すべき価値関数(value function)

であり，それら一連の動作をまとめると以下のようになる。

各期の初め(以下期首と呼ぶ)にエージェントには，環境の状態とエージェントの決定に基づいた報酬が与えられる。その報酬に基づき，エージェントは状態に対する価値関数を更新し，将来における価値関数が最大になるように決定を定める。この動作の繰り返しによりエージェントはより良い（最適な）行動規律を獲得していく。

強化学習において，エージェントは予め環境の状態に対する評価を持たず，環境の状態推移とそれに伴う報酬は確率的に与えられる。エージェントは，価値関数を最大化するために，現在までに得られた情報と過去に行った決定から最良の決定を選択する。しかし，そのような決定を発見するには，現在までに経験していない決定を選択する試みを積極的にとりいれなければならない。すなわち，これまでに得られた情報を利用しながら，将来において最良の決定を実現するために未知の決定を探検しなければならない。この探検が少なすぎると，未知の決定に対する学習が行われず，将来のより良い決定を発見できなくなり，また逆に探検が多すぎると，よい決定が得られているにもかかわらず，無駄な探検を続けることになる（過学習）。得られた情報の利用と探検は，トレードオフの関係にあり，学習の初期の段階では，情報が少ないために探検を多くし，学習の時間経過に伴って情報が多くなると，探検を少なくしていくことが一般的である。また，価値は「あるものがどの程度役に立つか」という指標であり，決定から生じうる様々な結果の平均が価値関数である。価値と決定が引き起こす結果の確率を結合すれば，それぞれの決定の価値関数が得られる。

以上の強化学習モデルは，第 n 期首の状態 s_n とその時の決定 a_n により，n 期の利得 r_n と第 $n+1$ 期首の状態 s_{n+1} がマルコフ的に与えられる MDP として定式化できる。すなわち，

h_n : n 期までの履歴（$s_0, a_0, r_0, \cdots, s_n$）

とおけば，推移確率と平均利得は

$$p(s,s',a)=P\{s_{n+1}=s'|h_{n-1},s_n=s,a_n=a\}$$
$$=P\{s_{n+1}=s'|s_n=s,a_n=a\}$$
$$r(s,a)=E\{r_n|h_{n-1},s_n=s,a_n=a\}$$
$$=E\{r_n|s_n=s,a_n=a\}=\sum_{s'\in S}p(s,s',a)r(s,s',a)$$

で与えられる。本論文を通じて，費用最小化問題を取り扱うため，以下利得を費用と読み替える。

g を 1 期当たりの最小平均費用，$h(s)$ を相対費用とおけば，無限期間にわたる時間平均利得を最小化する最適性方程式は，対象とする単一連鎖にたいして

$$g+h(s)=\min_{\mathbf{a}\in K(s)}\left\{r(s,\mathbf{a})+\sum_{s'\in S}p(s,s',\mathbf{a})h(s')\right\},s\in S\ (1)$$

となる。ここで，S は状態集合であり，

$K(s)$：状態 s でとりうる決定の集合

である。最適政策は，各 s で(1)式右辺を最小化する決定として定められる。ここで，相対費用 $h(s)$ は適当に定められた状態 s_r で $h(s_r)=0$ である[1,2]。

3. SMART と SBPI アルゴリズム

既存のニューロ DP アルゴリズムとして SMART[12]を紹介する。SMART は，TD 法を用いて，シミュレーションをもとに Q-学習を行うアルゴリズムである。ここで，Q-学習における Q 値とは，各状態における各決定の効用値であり，状態と決定の関数である。反復のある時点における最小の Q 値は，それまでの学習から得られた情報の中での最小値である。そこで，学習を行っていない領域を学習させるために，ある確率で最小でない Q 値を持つ決定を選択する。このステップを探検とよび，以下のアルゴリズムのステップ 3 である。なお，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。

[SMART] [12]

ステップ 1：全ての $s\in S$ と $\mathbf{a}\in K(s)$ にたいして Q-factor $Q_{new}(s,\mathbf{a})=Q_{old}(s,\mathbf{a})=0$，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，学習率と探検確率をステップ 2 で定めるパラメータ $(\alpha_0,\alpha_\tau,p_0,p_\tau)$ を与える。

ステップ 2：反復 k で状態 s 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 $Q_{new}(s,\mathbf{a})$ を最小にする決定 $\mathbf{a}^*$ を選択し，確率 p_k で $\mathbf{a}^*$ を除く $K(s)$ からランダムに $\mathbf{a}$ を選択する。

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

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

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

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

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

ステップ 7：$Q_{old}(s,\mathbf{a})=Q_{new}(s,\mathbf{a})$ 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 $k=k+1$，s' を s としてステップ 2 へ。

Gosavi[13]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。

[RELAXED–SMART] [13]

ステップ 3～5，7，8 は[SMART]と同じである。

ステップ 1：Q-factor $Q_{new}(s,\mathbf{a})=Q_{old}(s,\mathbf{a})=0$，$TC=0$，$T=0$，$g=0$，$k=0$ 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sにいれば，α_k，p_k，β_kを
$\alpha_k=\alpha_0/k$，$p_k=p_0/k$，$\beta_k=\beta_0/k$
として定める。
ステップ6：ステップ3で決定$\mathbf{a}^*$を選択したならば，TC，T，gを次式で更新する。
$$TC=(1-\beta_k)TC+\beta_k r(s,s',\mathbf{a}^*)$$
$$T=(1-\beta_k)T+\beta_k$$
$$g=TC/T$$

一方，He et al.[14]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] [14]
ステップ1：初期政策$\{f^0(s);s\in S\}$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s_0からシミュレーションにより$s_1,\cdots,s_m$を生成する。
ii) $g^k=0$とおき，$n=0,\cdots,m-1$にたいして(s_n,s_{n+1})の推移に伴うg^kを次式で更新する。
$$g^k=(1-1/(n+1))g^k+(1/(n+1))r(s_n,s_{n+1},f^k(s_n))$$
2-b：($h^k(s)$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s^*とおく。
ii) 過渡状態s_0から出発し，状態s^*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) l本目のトラジェクトリ$(s_0,s_1,\cdots,s_N=s^*)$，$l=1,\cdots,L$，にたいして，推移(s_n,s_{n+1})に伴う$w(s_i)$，$i=1,\cdots,n$，を次式により更新する。
$$w(s_i)=w(s_i)+\gamma_i\lambda^{n-i}d_n$$

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，
$0\le\lambda\le1$，
$$d_n=r(s_n,s_{n+1},f^k(s_n))-g^k+w(s_{n+1})-w(s_n)$$
である。
iv) $h^k(s)=w(s)-w(s_r)$，$s\in S$
ステップ3：(政策改良ルーチン)
$$f^{k+1}(s)=\arg\min_{\mathbf{a}\in K(s)}\left\{r(s,\mathbf{a})+\sum_{s'\in S}p(s,s',\mathbf{a})h^k(s')\right\},\quad s\in S$$
ステップ4：$f^{k+1}(s)=f^k(s)$，$s\in S$ならば停止。最適政策は$f^k(s)$である。さもなければ$k=k+1$としてステップ2へ。

4. MPIMとSBMPIMアルゴリズム

最適性方程式(1)を解くアルゴリズムが政策反復法（PIM）であり，修正政策反復法(MPIM)である[1-5]。特に，MPIMはPIMの値決定ルーチンを有限回の反復で置き換えた手法であり，比較的規模の大きな問題に対しても有効である。
[MPIM] [5]
ステップ1：$h_0(s_r)=0$をみたす初期ベクトルh^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$s_n\in S$に対して，
$$g^{k+1}(s)=\min_{\mathbf{a}\in K(s)}\left\{r(s,\mathbf{a})+\sum_{s'\in S}p(s,s',\mathbf{a})h^k(s')-h^k(s)\right\}$$
を計算し，$f^k(s_n)$が$g^{k+1}(s)$を与えれば，$f^{k+1}(s)=f^k(s)$とおき，さもなければ，$g^{k+1}(s)$を与える任意の決定を$f^{k+1}(s)$ととる。
ステップ3：(値近似ルーチン)
$$w^0(s_n)=h^k(s_n)+g^{k+1}(s_n),\quad s_n\in S$$
とおき，$l=0,1,\cdots m-1$に対して順次，
$$w^{l+1}(s)=r(s,f^{k+1}(s))+\sum_{s'\in S}p(s,s',f^{k+1}(s))w^l(s'),\quad s\in\mathbf{S}$$
を計算し，
$$h^{k+1}(s)=w^m(s)-w^m(s_r),\quad s\in S$$
とおく。すべてのsに対して，

$|h^{k+1}(s_n)-h^k(s_n)|<\varepsilon$ であれば終了。さもなければ，$k=k+1$ として，ステップ 2 へ。

しかし，生産ラインの工程数が増加すると，状態空間 S 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 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 にたいしてだけ相対費用 $h(s)$ 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。

[SBMPIM] [19]

ステップ 1：初期状態 s_0 と望ましい状態 s^*，収束判定のための状態の集合 S_0 を定め，シミュレーション回数 m，非負数 λ（$\lambda\le 1$）および停止基準の正整数 Q と $\varepsilon>0$ を定めて，訪問した状態の集合 $S_v=\phi$（空集合），$S_T=\phi$，累積費用 $TC=0$，$s=s_0$，$k=l=1$，$q=0$ とおく。

ステップ 2：（Schweitzer 変換[29]）次式を満たす正数 τ

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

を定め，直接費用 $r(s,\mathbf{a})$，推移確率 $p(s,s',\mathbf{a})$ を以下の式で変換する。

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

ここで $\delta_{s,s'}=1$，$s=s'$；　$=0$，$s\neq s'$ である。

ステップ 3：$s\notin S_v$ 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る。$s\in S_v$ 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。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' を定める。

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

と更新し，$l=m$ 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 へ。

ステップ 4：（g の推定）1 期当たりの平均費用 $g(k)$ および g を次式

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

により計算する。

ステップ 5：（$h(s)$ の推定）S_T のなかで $v(s)$ が最大の s を s_r とおき，

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

を計算し，$s(\neq s_r)\in S_v$ にたいして

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

を計算し，$h(s_r)=0$ とおく。ただし，$k=1$ のときには

$$h(s_r)=r(s_r,f(s_r))-g,$$
$$h(s)=r(s,f(s))-g-h(s_r)$$

である。

ステップ 6：（政策改良ルーチン）$s\in S_v$ にたいして

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

を計算する。ここで $N(s,f(s))$ は $K(s)$ における $f(s)$ の近傍であり，$p(s,s',\mathbf{a})>0$ となる $s'\notin S_v$ にたいしては，$S_v=S_v\cup\{s'\}$ とおき，$f(s')$ を s^* へ向かう実行可能な決定と定める。$w(s')=r(s',f(s'))$ とおき，

$$h(s')=r(s',f(s'))-r(s_r,f(s_r))$$

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。全ての $s\in S_0$ で $f(s)$ が改良されなければ　$q=q+1$ とおき，さもなければ $q=0$ とおく。$q\ge Q$ ならば，$\{g(k-q)/\tau,\cdots,g(k)/\tau\}$ 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，

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

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 $S_T=\phi$，$TC=0$，

$l=1$, $k=k+1$とおきステップ3へ。

5. ニューロDPアルゴリズムの比較

単一品種単一工程生産ラインの最適制御問題を対象に3,4章で述べたニューロDPアルゴリズムの比較を行う。次章では,3工程生産・物流システムを対象に,準最適を基準にしたプル方式間の比較を行うので,ここで一般的な単一品種生産・物流システムの最適制御問題を簡単に紹介する。

第1工程が外注工場等から部品を購入し,単一品種の製品を完成させるM工程生産・物流システムを考える。各工程 i, $i=1,\cdots,M$, の発注,発送,納入は各期首に行われ,前工程は輸送時間T_iを含む一定の納入リードタイムL_i ($>T_i$) 期後に受注した部品を納入する。ただし,必要な量の部品がなければ,不足分は次期に繰り越されるものとする。工程 i の部品の最大在庫量を$I_{\max;i}$,製品の倉庫容量を$J_{\max;i}$,公称の生産能力をC_iとおく。しかし,機械故障等のためC_iは達成できず,n, $n=1,2,\cdots$, 期における生産能力$C_i(n)$は,各期独立に同一の離散分布に従うものとし,その最小値を$C_{i;\min}$とする。同様に,最終製品にたいするn期の需要量$D(n)$も,互いに独立で同一の離散分布に従うものとし,その最小値と最大値を$D_{\min}$, $D_{\max}$とし平均をDとおく。満たされなかった需要は注文残となるが,システムの状態数を有限にするため最大$B_{\max}$まで受注されるものとし,それを越えた需要は失われるものとする。

第i工程は,第n期首において部品在庫量$I_i(n)$と製品在庫量$J_i(n)$を持つものとし,それらシステム全体の情報に基づいて,その期の部品発注量$O_i(n)$,製品生産量$P_i(n)$を決定する。$J_i(n)$の負の値は工程$(i+1)$の発注残(品切れ)を意味している。そして,n期首における工程$i-1$からiへの発送量を$Q_i(n)$とおく。

この生産・物流システムにたいして,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。費用としては,輸送中を含む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。すなわち,$i=1,\cdots,M$にたいして

C_i^I :各期における工程iの部品在庫費用/個

C_i^J :各期における工程$i+1$への輸送中を含む工程iの製品在庫費用/個

C_i^B :各期における工程iの発注残費用/個

B_i :各期における工程iの発注残発生費用/回

である。また,最終工程で$B_{\max}$をこえて失われた需要にたいして損失費用$C_{\max}$/個がかかるものとする。

この最適制御問題をMPIMで解くことを考える。簡単のため,$i=1,\cdots,M$にたいして$L_i=1$, $I_{\max;i}=I_{\max}$, $J_{\max;i}=J_{\max}$とおけば,状態空間Sの要素数は

$$(I_{\max}+1)^M(J_{\max}+I_{\max}+1)^{M-1}(J_{\max}+B_{\max}+1) \tag{2}$$

である。例えば,$I_{\max}=J_{\max}=B_{\max}=9$のとき$190^M$である。

M=1とおいた単一工程生産ラインにたいしては,(2)式に示されるように状態数は通常1000以下であり,4章のMPIMを適用することができる。そこで,この厳密解をもとに,3,4章で紹介したニューロDPアルゴリズムSBMPIM, SMART, RELAXED-SMART, SBPIの数値比較を行う。

パラメータを以下のように設定する。

$M=L=1$, $I_{\max}=8$, $J_{\max}=5$, $B_{\max}=0$, $C_{\max}=100$, $C=5$, $C_{\min}=3$, $C^I=1$, $C^J=2$, $C^B=5$, $B=10$

したがって,状態数は144である。そして,工程故障を考慮した生産能力分布$P(C(n)=c)=P_c$, $C_{\min}\le c\le C$として

$P_5=0.7$, $P_4=0.2$, $P_3=0.1$

とし,需要$D(n), n=1,2,...$の分布は,変形した二項分布

$$\Pr\left\{D(n)=D-\frac{1}{2}Q+j\right\}=\binom{Q}{j}\left(\frac{1}{2}\right)^{Q},\ 0\le j\le Q \tag{3}$$

に従うものとする。ここで，D は整数，Q は偶数($Q\le 2D$)であり，分布の平均は D，分散は $Q/4$ である。本章の数値比較においては，D=3，Q=2を用いる。

各アルゴリズムにおけるパラメータは，MPIM にたいして $m=20$，$\varepsilon=10^{-5}$ であり，SBMPIM において $m=10^3$，$\lambda=0.15$，$s_0=(0,-10)$，$s^*=(8,5)$，停止回数$=10^5$，ステップ 5 の $N(s,f(s))=K(s)$，SMART にたいして $(\alpha_0,\alpha_\tau,p_0,p_\tau)=(0.1,5\times10^5,0.1,5\times10^5)$，停止回数$=5\times10^5$，RELAXED–SMART にたいして $(\alpha_0,p_0,\beta_0)=(0.1,0.1,0.9)$，停止回数$5\times10^5$，SBPI にたいして $s_0=(4,0)$，$m=10^3$，$\lambda=0.9$ とした。最近，Gosavi ら[15]は，SMART の $Q(s,a)$ の更新を SBPI 同様ある状態 s から s へのトラジェクトリに基づく方式に変更し，λ-SMART と呼んでいる。λ-SMART のパラメータは $s_0=(3,-10)$，シミュレーションの長さ$=5\times10^5$，$(\alpha_0,\alpha_\tau,p_0,p_\tau,\lambda)$ =(0.9, 5×10^5, 0.99, 5×10^5, 0.9)とした。

まず，上記パラメータ設定のもとでの各アルゴリズムの計算時間および最終の平均費用 g を表 1 に示す。ここで MPIM の反復回数は 5 回であった。計算機は DOS/V 機(CPU:Pentium4 2.4GHz，メモリ:2GB)を用い，プログラミング言語は容易に実行できる ExcelVBA を用いた。表 1 より，SBMPIM 以外のニューロ DP アルゴリズムは，MPIM の最小平均費用に収束していないことがわかる。

次いで，これらニューロ DP アルゴリズムの収束状況を示すために，1 秒ごとの平均費用の推移を示したものが表 2 である。さらに表 3 は，MPIM による最適政策のもとでの再帰状態近辺の各アルゴリズムの最終政策をまとめたものである。MPIM の列に最適政策が示されており，第 1 列が発注量，第 2 列が生産量を示している。また，最適政策のもとでの再帰状態が*で示されている。各ニューロ DP アルゴリズムにおける第 3 列の○は最適政策と一致したことを示している。これらから SBMPIM が他のニューロ DP アルゴリズムより優れていることは明らかである。

JIT 生産システムでは，かんばん方式が日常的に用いられている。そこで，同じ単一工程をかんばん方式で運用した際の最小平均費用をシミュレーションにより求めた。シミュレーションには，乱数発生にメルセンヌ・ツイスターを用いた共通乱数法（[30]，第 9 章）を分散減少法として採用し，バッチサイズ10^4，バッチ数 20 のバッチ平均法を用いた。バッチ平均法による結果は，引き取りかんばん 7 枚，生産指示かんばん 4 枚のとき，平均費用が最小となり，6.403±0.036 であった。表 1 のニューロ DP アルゴリズムの性能をはるかに上回るものである。ただ，MPIM による最適制御にくらべると，費用は約 7%増加する。そして，再帰状態(3, 0), (4, 0)で最適政策と一致するものの，(3, 1)では発注量 4，生産量 3，(3, 2)では発注量 4，生産量 2，(4, -1)では発注量 3，生産量 4，(4, 1)では発注量 3，生産量 3 と状態(4, -1)を除いていずれも最適発注量を上まわっている。

6. プル方式間の比較

3 工程生産・物流システムを対象に，各プル方式を最適設定で運用したときの最小平均費用をシミュレーションにより求め，SBMPIM による準最小平均費用と比較し，各プル方式がどれだけ準最適に近いかを明らかにする。対象とするプル方式は，かんばん方式，基点在庫方式，CONWIP，ハイブリッド方式，一般化かんばん方式の 5 方式であり，その違いが図 1 に示されている。以下，各プル方式の下での発注指示量と生産指示量を示し，次期の状態への推移を表す関係式をまとめておく。

1) かんばん方式[21]
工程 i, $i=1,2,3$, の引き取りかんばん枚数を M_i, 生産指示かんばん枚数を N_i で表せば, 初期状態は $I_i(1)=M_i$, $J_i(1)=N_i$ であり,

$$O_i(n)=M_i-I_i(n)-[-J_{i-1}(n)]^+-\sum_{l=0}^{T_i-1}Q_i(n-l)$$

$$P_i(n)=\min\left\{N_i-[J_i(n)]^+, I_i(n), C_i\right\}$$

である。このとき, 次期首の状態は

$$I_i(n+1)=I_i(n)+Q_i(n-T_i+1)-P_i'(n) \quad (4)$$

$$J_i(n+1)=J_i(n)+P_i'(n)-O_{i+1}(n-L_{i+1}+T_{i+1}+1),\ i=1,2 \quad (5)$$

$$J_3(n+1)=\max\{J_3(n)+P_3'(n)-D(n), -B_{\max}\} \quad (6)$$

である。ここで,

$$P_i'(n)=\min\{P_i(n), C_i(n)\} \quad (7)$$

$$Q_1(n)=O_1(n-L_1+T_1) \quad (8)$$

$$Q_i(n)=\min\left\{O_i(n-L_i+T_i)+[-J_{i-1}(n-1)]^+, P_{i-1}'(n-1)+[J_{i-1}(n-1)]^+\right\},\ i=2,3 \quad (9)$$

である。

2) 基点在庫方式[22,23]
工程 i, $i=1,2,3$, にたいする基点在庫量を S_i とおけば,

$$I_i(1)=0,\quad J_i(1)=S_i$$

であり, $n=1,2,\cdots$ にたいして

$$O_i(n)=D(n-1),$$

$$P_i(n)=\min\left\{S_i-[J_i(n)]^+, I_i(n), C_i\right\}$$

である。ここで $D(0)=0$ である。次期首の状態等は式(4)－(9)で定められる。

3) CONWIP[24]
総 WIP を S とおけば

$$S=\sum_{i=1}^{3}(I_i(1)+J_i(1))$$

であり, $n=1,2,\cdots$ にたいして

$$O_1(n)=D(n-1),\quad O_i(n)=J_{i-1}(n),\quad i=2,3$$

$$P_i(n)=\min\left\{I_i(n), C_i\right\},\quad i=1,2,3$$

である。次期首の状態等は式(4)－(9)で定められる。

4) ハイブリッド方式[25]
総 WIP を S, 工程 $i=2,3$ の引き取りかんばん枚数, 生産指示かんばん枚数を M_i, N_i とおけば

$$I_1(1)=0,\quad J_1(1)=S-\sum_{i=2}^{3}(I_i(1)+J_i(1)),\quad I_i(1)=M_i,$$

$$J_i(1)=N_i,\quad i=2,3$$

であり,

$$O_1(n)=D(n-1),\quad P_1(n)=\min\{I_1(n), C_1\}$$

$$O_i(n)=M_i-I_i(n)-[-J_{i-1}(n)]^+-\sum_{l=0}^{T_i-1}Q_i(n-l)$$

$$P_i(n)=\min\left\{N_i-[J_i(n)]^+, I_i(n), C_i\right\},\quad i=2,3$$

である。次期首の状態等は式(4)－(9)で定められる。

5) 一般化かんばん方式[26-28]
工程 i, $i=1,2,3$ にたいする基点在庫量を S_i, 引き取りかんばん枚数を M_i で表せば,

$$I_i(1)=M_i,\quad J_i(1)=S_i$$

であり, $n=1,2,\cdots$ にたいして

$$O_i(n)=\min\left\{D(n-1), M_i-I_i(n)-[-J_{i-1}(n)]^+-\sum_{l=0}^{T_i-1}Q_i(n-l)\right\}$$

$$P_i(n)=\min\left\{S_i-[J_i(n)]^+, I_i(n), C_i\right\}$$

である。次期首の状態等は式(4)－(9)で定められる。

以上のプル方式間の比較を 3 工程生産・物流システムの最適制御問題を対象に, SBMPIM により計算された準最小費用を基準にして行う。

パラメータを *i=1,2,3* にたいして以下のように設定する。
$L_i=1$, $T_i=0$, $I_{\max:i}=8$, $J_{\max:i}=5$, $B_{\max}=4$
したがって, (2)式より状態数は 143 万である。

まず, 需要は単一工程と同じく, (3)式で与えられる分布に従うものとし, D=Q=2 とおく。平均 2, 分散 0.5 である。
また, 各工程の生産能力分布として,
A : $P_5=1$ (故障なし), 平均生産能力＝5
B : $P_5=0.7$, $P_4=0.2$, $P_3=0.1$, 平均生産能力＝4.6
C : $P_5=0.4$, $P_4=0.3$, $P_3=0.2$, $P_2=0.1$, 平均生産能力＝4.0
の 3 分布を考えることにする。最後に,

費用構造としては,

(a)

$(C_1^I, C_2^I, C_2^I) = (1,3,6)$,

$(C_1^J, C_2^J, C_3^J) = (3,6,12)$,

$(C_1^B, C_2^B, C_3^B) = (20,40,80)$,

$(B_1, B_2, B_3) = (40,80,120)$, $C_{\max} = 1000$

(b)

$(C_1^I, C_2^I, C_3^I) = (1,2,4)$,

$(C_1^J, C_2^J, C_3^J) = (2,4,6)$

$(C_1^B, C_2^B, C_3^B) = (10,10,30)$,

$(B_1, B_2, B_3) = (20,20,60)$, $C_{\max} = 100$

の2ケースを考える。

この生産・物流システムの最適制御問題を，SBMPIMのパラメータを

$s_0 = (3,3,3,3,3,5)$, $s^* = (8,5,8,5,8,5)$,

$m = 10^5$, $\lambda = 0.99$, $\tau = 0.99$, $Q = 20$,

$\varepsilon = 0.1$

と設定して計算し，得られた最小平均費用を表**4-(a)**および**(b)**のSBMPIMの行に示している。

プル方式間の比較をなるべく公平に行うことを考える。まず，プル方式の評価にはシミュレーションを用いるため，予め$B_{\max}$を設定する必要がなく，したがって$C_{\max}$を導入する必要もない。また，品切れ費用も顧客にたいしてだけ考慮することとし，$C_1^B = C_2^B = B_1 = B_2 = 0$と設定する。これら以外は上記パラメータの値を採用する。そして，各プル方式の単位期間当たりの平均費用を最小化する最適設定は，5章のかんばん方式同様のシミュレーションにより行う。

得られた各方式の最小平均費用が表**4-(a)**および**(b)**の各方式の行に示されており，各方式のSBMPIMに比べた平均費用の増加率（%）を次式

（各方式の信頼区間の最小値－SBMPIMの信頼区間の最大値）／(SBMPIMの信頼区間の最大値) (%)

で求めた値および各方式の最適パラメータの値が（ ）内に示されている。表**4**よりプル方式間では，5), 2), 1), 4), 3)の順に優れていることがわかる。しかし最良の一般化かんばん方式においても，**(a)**のケースでSBMPIMの準最適政策を採用することで，A～Cにたいして各々平均費用を少なくとも24.13%，18.62%，5.30%ずつ低減させることができる。[31]では，MRP，かんばん方式，TOC，CONWIP等を含む代表的な生産管理方式とプル方式を説明し，かんばん方式，MRP，プル方式間の比較および生産ラインの最適制御に関する従来の研究を概観し，プル方式がWIPを制御するのにたいし，プッシュ方式は生産率（throughput）を制御するため，プル方式の方が制御しやすく，制御量の最適値からのずれに頑健で，混雑しにくいことが指摘されている。実際，表**4-(a)**および**(b)**を通して，各方式の最適パラメータの値はほとんど変化せず，プル方式の頑健性が実証されている。

現在急速に研究が進展しているサプライチェーンマネージメント(Supply Chain Management，SCM)への適用を考慮すると，輸送時間を取り入れなければならない。簡単のため，最終工程への輸送時間を1にとる。すなわち，納入リードタイム L_3=2, T_3=1 である。需要分布として D=Q=2, 各工程の生産能力分布としてAをとり，費用構造として

$(C_1^I, C_2^I, C_3^I) = (1,3,6)$

$(C_1^J, C_2^J, C_3^J) = (3,6,12)$

$(C_1^B, C_2^B, C_3^B) = (0,0,80)$,

$(B_1, B_2, B_3) = (0,0,120)$,

$C_{\max} = 1000$

とおく。平均需要量2を処理するためには，各工程のパラメータは i=1,2 にたいして,

$L_i = 1$, $T_i = 0$, $I_{\max i} = 6$, $J_{\max i} = 6$

ととり,

$L_3 = 2$, $T_3 = 1$, $I_{\max:3} = 10$, $J_{\max:3} = 10$,

$B_{\max} = 5$

ととる必要がある。このとき，輸送量を状態変数として考慮しなければならず，(2)式同様に計算すれば，状態数は 2096 万になる。このため，SBMPIM を DOS/V 機(OS:Windows XP x64, CPU:インテル Core 2 Duo 2.66GHz, メモリ:4GB)で実行したが，メモリーオーバーで計算できなかった。この点を改良したものを SBMPIM Ver.2 として以下に示す。

[SBMPIM Ver.2]

ステップ 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，非負数 λ, μ ($\lambda, \mu \le 1$)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。

ステップ 2：(Schweitzer 変換[29])

次式を満たす正数 τ

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

を定め，直接費用 $r(s,\mathbf{a})$，推移確率 $p(s,s',\mathbf{a})$ を以下の式で変換する。

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

ここで $\delta_{s,s'} = 1, s = s'$；$= 0, s \neq s'$ である。

ステップ 3：(シミュレーション)

i) 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，

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

と更新する。

ii-1) $s \notin S_V$ 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。

ii-2) $s \notin S_V$ 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。

ii-3) $s \in S_v$ 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。

iii) $l = m$ 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-i)へ。

ステップ 4：(g の推定)

S_T のなかで $v(s)$ が最大の s を s_r とおき，1 期当たりの平均費用 $g(k)$ および g を次式

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

により計算する。

ステップ 5：($h(s)$ の推定)

$h(s_0) = (1 - \lambda^k v(s_0)/m) w(s_0) + (\lambda^k v(s_0)/m) r(s_0, f(s_0)) - g$

を計算する。

i) $s (\neq s_0) \in S_T$ にたいして

$h(s) = (1 - \lambda^k v(s)/m) w(s) + (\lambda^k v(s)/m) r(s, f(s)) - g - h(s_0)$

とおく。

ii) $s \in (S_v \cup S_U) - S_T$ にたいして

$$h(s) = w(s) - g - h(s_0)$$

とおく。

iii) $h(s_0) = 0$ とおく。

ステップ 6：(政策改良ルーチン)

i) $s \in S_v$ にたいして

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

を計算する。ここで $N(s, f(s))$ は $K(s)$ における $f(s)$ の近傍であり，$p(s,s',\mathbf{a}) > 0$ となる $s' \notin S_v \cup S_U$ にたいしては，$S_U = S_U \cup \{s'\}$ とおき，$f(s')$ を s^* へ向かう実行可能な決定と定め，

$$w(s') = r(s', f(s')),$$
$$h(s') = r(s', f(s')) - r(s_0, f(s_0))$$

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。

ii) $s \in S_U$ にたいして

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

を計算する。ここで $N(s, f(s))$ は $K(s)$ における $f(s)$ の近傍であり，$p(s,s',\mathbf{a}) > 0$ とな

る$s' \notin S_v \cup S_U$にたいしては,

$$h(s')=r(s',f(s'))-r(s_0,f(s_0))$$

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。

ステップ7：(収束判定)

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

ii) $q=1$ならば$s_0=s_r$とおき，ステップ iv)へ。さもなければ，$q<Q$のとき，ステップ iv)へ行き，$q \geq Q$ならば，ステップ iii)へ。

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

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

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 iv)へ。

iv) s=s$_0$, $S_T=\phi$, $TC=0$, $l=1$, $k=k+1$とおきステップ3へ。

以上において，ステップ1，3等における，状態s^*へ向かう実行可能な決定$f(s)$は，$s=(I_1,J_1,I_2,J_2,I_3,J_3,Q_3)$，$s^*=(I_1^*,J_1^*,I_2^*,J_2^*,I_3^*,J_3^*,Q_3^*)$にたいして

$$O_i=[I_i^*-I_i]^+, i=1,2,\ O_3=[I_3^*-I_3-Q_3^*]^+$$

$$P_i=\min\{I_i,C_i,J_i^*-J_i\}, i=1,2,3$$

として計算されている。

[SBMPIM Ver.2]による上述のパラメータのもとでの計算結果が，表 **5** の *D=2* に示されている。なお，参考までに示されている *D=1* は，需要分布 *D=1,Q=2* にたいするものである。

7. おわりに

本論文では，既存のニューロ DP アルゴリズムとして SMART[12,13] と SBPI[14]を紹介し，単一品種単一工程生産ラインの最適制御問題を対象に，これらニューロ DP アルゴリズムと提案した SBMPIM の比較を行った。さらに，[19,20] では考慮していなかった一般化かんばん方式をも取り入れたプル方式間の比較を，SBMPIM による準最適政策を基準にして行った。今後の課題としては，部品在庫，製品在庫の持ち方と納入リードタイムを再検討し，一般かんばん方式より優れた新方式の提案を行うことである。さらに，実用規模の問題に対しても適用できるニューロ DP アルゴリズムを開発するために，SBMPIM Ver.2 に基づき，分解法を適用するとともに，ニューラルネットワークを組み込み，各状態における最適政策を学習させる必要がある。

謝　辞

本研究の計算プログラムは全て名古屋工業大学伊藤崇博技官によるものであり，深謝する。またこの研究の一部は，愛知工業大学　研究特別助成および科学研究費補助金　基盤研究（C）１８５１０１３７の補助を受けたものである。

参　考　文　献

[1] Howard, R. A.:「Dynamic Programming and Markov Processes」, Cambridge, 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] Puterman, M. L.:「Markov Decision Processes」, John Wiley& Sons, (1994)

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

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

[5] Ohno, K.: "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)
[6] Ohno, K., Ishigaki, T. and Yoshii, T. : "A New Algorithm for a Multi-item Periodic Review Inventory System", *ZOR-Math. Methods of Oper. Res.* Vol. 39, pp. 349-364 (1994)
[7] Russell, S. and Norvig, P. (古川訳): 「エージェント アプローチ 人工知能」, 共立出版 (1997)
[8] Sutton, R. S. and Barto, A. G: 「Reinforcement Learning」, MIT Press (1998) (三上, 皆川訳: 「強化学習」, 森北出版 (2000))
[9] 銅谷, 森本, 鮫島: "強化学習と最適制御", システム／制御／情報, Vol.45, No.4, pp.186-196 (2001)
[10] Bertsekas, D.P. and Tsitsiklis, J.N.: 「Neuro-Dynamic Programming」, Athena Scientific (1996)
[11] Roy, R. V.: "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)
[12] Das, T. K., Gosavi, A., Mahadevan, S. and Marchalleck, N.: "Solving semi-Markov decision problem using average reward reinforcement learning", *Management Science*, Vol. 45, No. 4, pp. 560-574 (1999)
[13] Gosavi, A.: Doctor Thesis, http://faculty.uscolo.edu/gosavi/thesis.html (1999)
[14] He, Y., Fu, M. C. and Marcus, S. I.: "A Simulation- based policy iteration algorithm for average cost unichain Markov decision processes," pp. 161-182, in Laguna, M. and Velarde, J. L. G eds., 「Computing Tools for Modeling, Optimization and Simulation」, Kluwer Academic (2000)
[15] Gosavi, A., Bandla, N. and Das, T. K.: "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)
[16] Gosavi, A.: 「Simulation-based Optimization : Parametric Optimization Techniques and Reinforcement Learning」, Kluwer Academic (2003)
[17] Chang, H. S. Fu, M. C., Hu, J. and Marcus, S. I.: 「Simulation-based Algorithms for Markov Decision Processes」, Springer-Verlag (2007)
[18] 大野, 八嶋, 伊藤: "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)
[19] 大野,伊藤: "ニューロ・ダイナミックプログラミングによる生産・物流システムの最適制御とプル方式の比較",日本経営工学会論文誌, Vol.55, No.4, pp.174-188 (2004)
[20] 大野: "生産・物流システムの最適制御とＪＩＴ", ジャストインタイム生産システム研究会編「ジャストインタイム生産システム」, 日刊工業新聞社, pp.351-377(2004)
[21] 門田:「トヨタ プロダクションシステム—その理論と体系」, ダイヤモンド社 (2006)
[22] Magee, J. F.: 「Production Planning and Inventory Control」, McGraw-Hill(1958)
[23] Clark, A. J. and Scarf, H.: "Optimal policies for multi-echelon inventory problem," *Management Science*, Vol. 6, pp. 475-490 (1960)
[24] Spearman, M. L., Woodruff, D. L. and Hopp, W. J.: "CONWIP: A pull alternative to Kanban," *Inter. J. Prod. Res.*, Vol. 28, pp. 879-894 (1990)
[25] Bonvik, A. M., Couch, C. E. and Gershwin, S. B.: "A comparison of production-line control mech- anisms", Inter. J. Prod. Res., 35, 789-804 (1997)
[26] Buzacott, J. A. and Shanthikumar, I. G: 「Stochastic Models of Manufacturing Systems」, NJ, Prentice Hall (1993)

[27] Karaesmen, F. and Dallery, Y.:"A performance comparison of pull type control mechanisms for multi-stage manufacturing," *Inter. J. Prod. Econ.*, Vol. 68, pp. 59-71 (2000)
[28] Zipkin, P. H.: 「Foundations of Inventory Management」, McGraw Hill (2000)
[29] Schweitzer, P.J.: "Iterative solution of the functional equations of undiscounted Markov renewal programming," *J. Math. Anal. Appl.* Vol. 34, pp.495-501 (1971)
[30] 大野, 田村, 森, 中島 :「生産管理システム」, 朝倉書店(2002)
[31] 大野 :"生産ラインの最適制御", 計測と制御, Vol. 42, No. 7, pp. 546-551 (2003)

表 1 計算時間と平均費用

解法	計算時間 (秒)	平均費用 (g)
MPIM	4.39	5.935
SBMPIM	9.66	5.909±0.067
SMART	14.97	212.000
RELAXED–SMART	15.41	215.209
λ −SMART	20.78	200.75
SBPI	0.47	76.753

表 2 平均費用 g の推移

秒	SBMPIM	SMART	RELAXED–SMART	λ − SMART	SBPI
1	5.83	211.99	214.55	0.00	76.75
2	5.88	212.00	214.88	263.00	
3	5.82	212.00	215.00	162.05	
4	5.70	212.00	215.07	239.00	
5	6.06	212.00	215.10	263.00	
6	5.81	212.00	215.13	159.64	
7	6.07	212.00	215.15	231.40	
8	5.87	212.00	215.16	170.46	
9	5.73	212.00	215.17	189.57	
10		212.00	215.18	173.38	
11		212.00	215.19	211.13	
12		212.00	215.19	160.86	
13		212.00	215.20	180.18	
14		212.00	215.20	147.13	
15				312.50	
16				192.83	
17				245.50	
18				312.50	
19				202.29	
20				195.92	

表3　各ニューロDPアルゴリズムによる最終政策の比較

再帰状態	部品在庫量	製品在庫量	MPIM		SBMPIM			SMART			RELAXED-SMART			λ−SMART			SBPI		
	3	-10	5	3	5	3	○	0	2		0	2		0	1		0	3	
	3	-9	5	3	5	3	○	0	0		0	0		1	3		0	3	
	3	-8	5	3	5	3	○	0	0		0	0		0	1		0	3	
	3	-7	5	3	5	3	○	0	0		0	0		0	0		0	3	
	3	-6	5	3	5	3	○	0	0		0	0		0	0		0	3	
	3	-5	5	3	5	3	○	0	0		0	0		0	0		0	3	
	3	-4	5	3	5	3	○	0	0		0	0		0	0		0	3	
	3	-3	5	3	5	3	○	0	0		0	0		0	0		0	3	
	3	-2	5	3	5	3	○	0	0		0	0		0	0		0	3	
	3	-1	5	3	5	3	○	0	0		0	0		0	0		0	3	
*	3	0	4	3	4	3	○	0	0		0	0		0	0		0	3	
*	3	1	3	3	3	3	○	0	0		0	0		0	0		0	3	
*	3	2	2	2	2	2	○	0	0		0	0		0	0		0	0	
	3	3	1	1	1	1	○	0	0		0	0		0	0		0	0	
	3	4	0	0	0	0	○	0	0	○	0	0	○	0	0	○	0	0	○
	3	5	0	0	0	0	○	0	0	○	0	0	○	0	0	○	0	0	○
	4	-10	4	4	4	4	○	0	2		0	2		3	4		0	4	
	4	-9	4	4	4	4	○	0	0		0	0		3	0		0	4	
	4	-8	4	4	4	4	○	0	0		0	0		0	0		0	4	
	4	-7	4	4	4	4	○	0	0		0	0		0	0		0	4	
	4	-6	4	4	4	4	○	0	0		0	0		0	0		0	4	
	4	-5	4	4	4	4	○	0	0		0	0		0	0		0	4	

	4	-4	4	4	4	4	○	0	0		0	0		0	0		0	4	
	4	-3	4	4	4	4	○	0	0		0	0		0	0		0	4	
	4	-2	4	4	4	4	○	0	0		0	0		0	0		0	4	
*	4	-1	4	4	4	4	○	0	0		0	0		0	0		0	4	
*	4	0	3	4	3	4	○	0	0		0	0		0	0		0	4	
*	4	1	2	3	2	3	○	0	0		0	0		0	0		0	0	
	4	2	1	2	1	2	○	0	0		0	0		0	0		0	0	
	4	3	0	1	0	1	○	0	0		0	0		0	0		0	0	
	4	4	0	0	0	0	○	0	0	○	0	0	○	0	0	○	0	0	○
	4	5	0	0	0	0	○	0	0	○	0	0	○	0	0	○	0	0	○

ここで，第1列の*はMPIMの最適政策のもとでのマルコフ連鎖の再帰状態を示し，MPIMと各NDPアルゴリズムにおける第1列は発注量を，第2列は生産量を表し，第3列の○は最適政策と政策が一致したことを示している。

表4 プル方式間の最小平均費用の比較（SBMPIMに比べた増加率（%））
（ ）内は各方式の最適パラメータ

(a)

生産能力分布	A		B		C	
SBMPIM	42.568±0.013		44.544±0.012		51.770±0.078	
5) 一般化かんばん方式	52.925±0.071 (6,3,6,3,6,6)	24.13%	52.925±0.071 (6,3,6,3,6,6)	18.62%	55.210±0.187 (8,3,9,3,8,6)	6.12%
2) 基点在庫方式	52.925±0.071 (3,3,6)	24.13%	52.925±0.071 (3,3,6)	18.62%	55.210±0.187 (3,3,6)	6.12%
1) かんばん方式	55.739±0.404 (5,3,5,3,5,3)	29.95%	55.739±0.404 (5,3,5,3,5,3)	24.19%	58.087±0.538 (5,3,5,3,5,3)	11.00%
4) ハイブリッド方式	70.072±0.430 (18,5,3,5,3)	63.55%	70.072±0.430 (18,5,3,5,3)	56.30%	73.132±0.638 (18,5,3,5,3)	39.82%
3) CONWIP	78.527±0.329 -15	83.65%	78.527±0.329 -15	75.50%	79.063±0.394 -15	51.73%

(b)

生産能力分布	A		B		C	
SBMPIM	28.002±0.007		28.002±0.007		30.410±0.038	
5) 一般化かんばん方式	31.964±0.034 (6,3,6,3,6,6)	12.30%	31.964±0.034 (6,3,6,3,6,6)	12.30%	32.719±0.068 (8,3,9,3,8,6)	7.20%
2) 基点在庫方式	31.964±0.034 (3,3,6)	12.30%	31.964±0.034 (3,3,6)	12.30%	32.719±0.068 (3,3,6)	7.20%
1) かんばん方式	35.058±0.174 (5,3,5,3,5,3)	19.70%	35.058±0.174 (5,3,5,3,5,3)	19.70%	35.975±0.227 (5,3,5,3,5,3)	14.80 %
4) ハイブリッド方式	44.282±0.190 (18,5,3,5,3)	36.50%	44.282±0.190 (18,5,3,5,3)	36.50%	45.489±0.265 (18,5,3,5,3)	32.70 %
3) CONWIP	46.004±0.312 -14	38.70%	46.004±0.312 -14	38.70%	46.286±0.176 -15	34.00 %
参考 MRP	13473.233±2099.569		13473.233±2099.569		13473.233±2099.569	

表5 プル方式間の最小平均費用の比較（納入リードタイム L_3=2, T_3=1）

平均需要量 D=1	平均費用	最適パラメータ	増加率(%)
SBMPIM	41.025±0.058		
一般化かんばん方式	52.366	4,2,4,2,6,5	27.465
基点在庫方式	52.366	2,2,5	27.465
かんばん方式	53.400	3,2,3,2,5,2	29.982
ハイブリッド方式	58.707	13,3,2,5,2	42.900
CONWIP	67.968	10	65.442

平均需要量 D=2			
SBMPIM	57.135±0.266		
一般化かんばん方式	68.366	6,3,6,3,9,8	19.102
基点在庫方式	68.366	3,3,8	19.102
かんばん方式	69.400	5,3,5,3,8,3	20.904
ハイブリッド方式	80.707	20,5,3,8,3	40.602
CONWIP	92.968	17	61.962

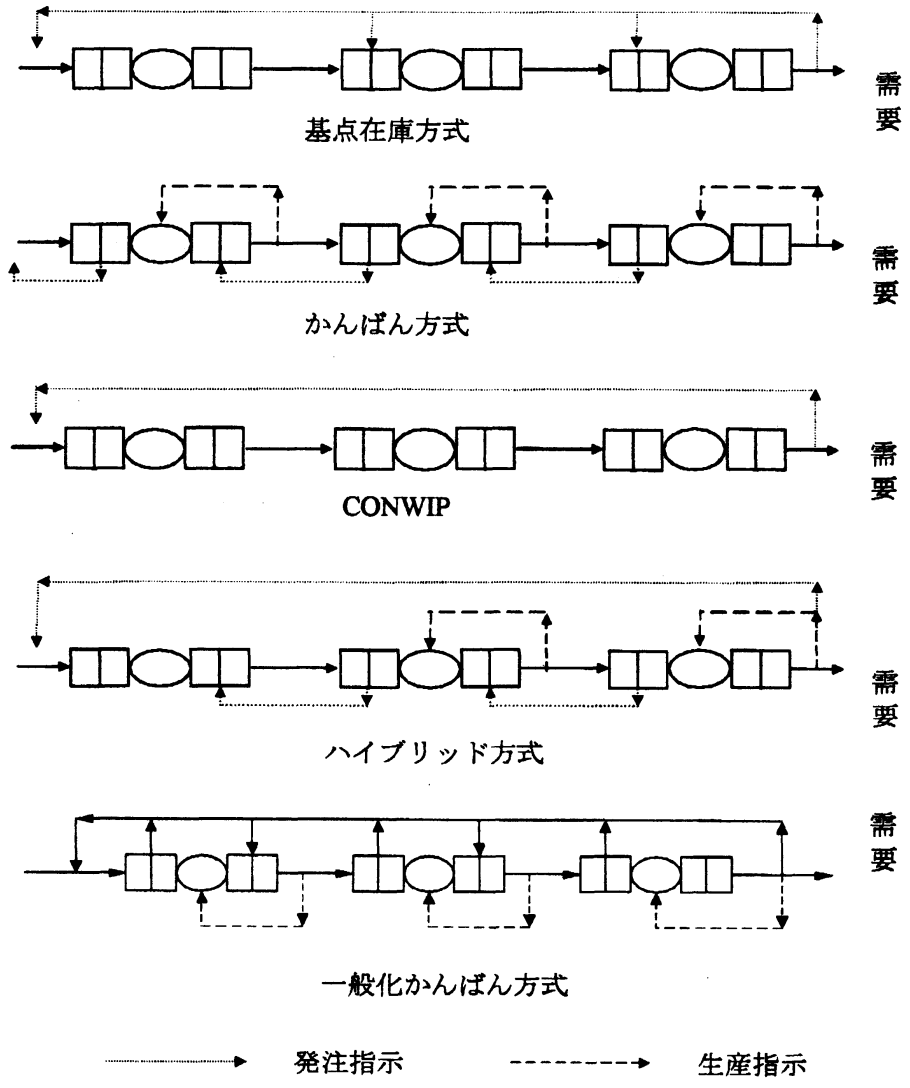
需要
基点在庫方式
需要
かんばん方式
需要
CONWIP
需要
ハイブリッド方式
需要
一般化かんばん方式
発注指示
生産指示

図1　プル方式