

## 最適かんばん方式とニューロDPによるSCMの最適化に関する研究

大野勝久\* (愛知工業大学経営学部)

坊敏隆, 荒川雅裕 (関西大学環境都市工学部)

### 1. はじめに

IT 革命の進展に伴い, 生産時点情報 (POP), 販売時点情報 (POS) 等の情報コストが劇的に低下している. 本研究は, ジャストインタイム (Just In Time, JIT) 生産システムに基づく多品目多段階サプライチェーンマネジメント (Supply Chain Management, SCM) の最適化を論ずる. すなわち, SC 全体の情報を利用した, 準最適な発注・生産・配送政策を求める, 次元の呪いを克服したニューロ DP (neuro-dynamic programming) アルゴリズムの開発を目標とする. 従来の研究[1-3]において, かんばん方式を含む 5 種類の最適化されたプル方式間の比較を, SBMPIM (Simulation-Based Modified Policy Iteration Method) による準最適政策を基準として行い, 拡張かんばん方式が最も優れた性能を持つことが示されている (5. の表 1 参照). また, これまでに提案されてきた代表的な強化学習アルゴリズムが, かんばん方式にも劣る政策しか生成できないことも明らかとなった.

本研究は当初, 性能の優れた拡張かんばん方式を最適化し, それをシミュレーションすることにより, SBMPIM に必要な情報を収集して, SBMPIM の高速化を目指していた. しかし, 拡張かんばん方式は JIT 生産システムと相性が良くないことが判明し, 最適化されたかんばん方式に変更して, SBMPIM を効率化し, 次元の呪いを克服せんとするものである. まず, 単一品種の製品を取り扱う, JIT 生産システムに基づくサプライチェーンの最適制御問題をマルコフ決定過程として定式化する.

### 2. JIT サプライチェーンの最適制御

図 1 に示される, 第 1 工程がサプライヤーから部品を購入し, 単一製品を加工して輸送・販売する  $M$  工程サプライチェーンを考える. 各工程  $i$  ( $=1, \dots, M$ ) の発注・生産指示は各期首に行われ, 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, 不足分は前工程の受注残 (品切れ) となるもの

とする. 生産工程  $i$  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,  $n$  ( $=1, 2, \dots$ ) 期における生産能力  $C_i(n)$  は, 各期独立な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. 輸送・販売工程についても同様に定義する. 生産能力同様, 最終製品に対する  $n$  期の需要量  $D(n)$  も, 互いに独立な同一の離散分布に従うものとし, その最小値と最大値を  $D_{\min}$ ,  $D_{\max}$ , 平均を  $D$  とおく. 満たされなかった需要は最終工程の受注残となるが, 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  $i$  の受注残を表している. そして,  $n$  期首における工程  $i-1$  から  $i$  への発注量を  $Q_i(n)$  とおく.

このサプライチェーンに対して, 単位期間あたりの平均総費用を最小化する最適発注・生産政策を決定する問題を考える. 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,  $i=1, \dots, M$  に対して以下の記号を定義する.

$C_i^I$ : 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  $i$  の受注残費用/個

$B_i$ : 各期における工程  $i$  の受注残発生費用/回

$C_{\max}$ : 最終工程で  $B_{\max}$  を超えて失われた需要に対する損失費用/個

第  $n$  期首における生産・物流システムの状態  $s_n$  は, 各工程  $i$  における第  $(n-L_i+T_i+1)$  期から第  $(n-1)$  期までの発注量, 工程  $i$  への第  $(n-T_i+1)$  期から第  $n$  期までの発注量, および部品在庫量と製品在庫量のベクトルによって表される. すな

わち,

$$s_n = (O_1(n-L_1+T_1+1), \dots, O_1(n-1), Q_1(n-T_1+1), \dots, Q_1(n), \dots, \\ O_i(n-L_i+T_i+1), \dots, O_i(n-1), Q_i(n-T_i+1), \dots, Q_i(n), \dots, \\ O_M(n-L_M+T_M+1), \dots, O_M(n-1), \\ Q_M(n-T_M+1), \dots, Q_M(n), \\ I_1(n), J_1(n), \dots, I_i(n), J_i(n), \dots, I_M(n), J_M(n)) \quad (2.1)$$

である。ここで  $L_i = 1$  の工程  $i$  にたいしては、その期の発注量が次期に納入されるため、発注量  $O_i$  の情報は不要である。したがって、全ての工程  $i$  で  $L_i = 1$  ならば

$$s_n = (I_1(n), J_1(n), \dots, I_i(n), J_i(n), \dots, I_M(n), J_M(n)) \quad (2.2)$$

である。これら可能なすべての状態  $s_n$  からなる状態空間を  $\mathbf{S}$  とおく。

状態  $s_n$  における工程  $i$  の可能な発注量  $O_i(n)$  と生産量  $P_i(n)$  の集合は、最大在庫量と生産能力の制限から各々次式で与えられる。

$$K_i^O(s_n) = \left\{ 0, \dots, I_{\max, i} - I_i(n) - \sum_{l=1}^{L_i - T_i - 1} O_i(n-l) - [ -J_{i-1}(n) ]^+ \right. \\ \left. - \sum_{l=0}^{T_i - 1} Q_i(n-l) \right\}, \quad i=1, \dots, M \quad (2.3)$$

$$K_i^P(s_n) = \{ 0, \dots, \min\{I_i(n), C_i, J_{\max, i} - J_i(n)\} \}, \quad i=1, \dots, M-1 \quad (2.4)$$

ここで、 $[x]^+ = \max(0, x)$ 、 $J_0(n) = 0$  である。最終工程  $M$  にたいしてはその後工程は市場であり、可能な生産量の集合は、最終製品の倉庫容量と需要の最小値を用いて次式で与えられる。

$$K_M^P(s_n) = \{ 0, \dots, \min\{J_M(n), C_M, J_{\max, M} - J_M(n) + D_{\min}\} \} \quad (2.5)$$

すなわち、状態  $s_n$  でとりうる決定

$$\mathbf{a} = (O_1(n), P_1(n), \dots, O_i(n), P_i(n), \dots, O_M(n), P_M(n))$$

は  $O_i(n) \in K_i^O(s_n)$ 、 $P_i(n) \in K_i^P(s_n)$ 、 $i=1, \dots, M$  を満たさなければならない。そして、式(2.3)~(2.5)で与えられる各工程の可能な発注量と生産量の集合の直積を  $K(s_n)$  で表すことにすれば、 $\mathbf{a} \in K(s_n)$  であり、政策  $f$  は、各状態  $s$  における可能な決定  $f(s)$  の集合  $\{f(s) \in K(s); s \in \mathbf{S}\}$  である。

政策が決定されれば、次の期首の状態は以下のように定められる。

$$I_i(n+1) = I_i(n) + Q_i(n-T_i+1) - P_i(n), \quad i=1, \dots, M \quad (2.6)$$

$$J_i(n+1) = J_i(n) + P_i(n) - O_{i+1}(n-L_{i+1}+T_{i+1}+1), \\ i=1, \dots, M-1 \quad (2.7)$$

$$J_M(n+1) = \max\{J_M(n) + P_M(n) - D(n), -B_{\max}\}, \quad (2.8)$$

ここで  $P_i'(n)$  は  $n$  期の実際生産量であり、

$$P_i'(n) = \min\{P_i(n), C_i(n)\} \quad (2.9)$$

で与えられる。また、各工程における発注量  $Q_i(n+1)$ 、 $i=1, \dots, M$ 、は次式で与えられる。

$$Q_i(n+1) = O_1(n-L_1+T_1+1) \\ Q_i(n+1) = \min\{O_i(n-L_i+T_i+1) + [ -J_{i-1}(n) ]^+, P_{i-1}(n) + [J_{i-1}(n)]^+\}, \\ i=2, \dots, M \quad (2.10)$$

そして、状態  $s_n$  で決定  $\mathbf{a}$  をとったとき、次に状態  $s_{n+1}$  へ推移する確率は、生産能力および需要量の分布を用いて以下のように与えられる。

$$p(s_n, s_{n+1}, \mathbf{a}) \\ \left\{ \begin{array}{l} \Pr\{C_i(n) = c_i, i=1, \dots, n, D(n) = d\}, \quad s_{n+1} = \\ \left\{ O_1(n-L_1+T_1+2), \dots, O_1(n), \right. \\ \left. Q_1(n-T_1+2), \dots, Q_1(n+1), \dots, \right. \\ \left. O_M(n-L_M+T_M+2), \dots, O_M(n), \right. \\ \left. Q_M(n-T_M+2), \dots, Q_M(n+1), \right. \\ \left. J_1(n) + Q_1(n-T_1+1) - \min\{P_1(n), c_1\}, \right. \\ \left. J_i(n) + \min\{P_i(n), c_i\} - O_2(n-L_2+T_2+1), \right. \\ \left. \dots, \quad I_M(n) + Q_M(n-T_M+1) - \min\{P_M(n), c_M\}, \right. \\ \left. \max\{J_M(n) + \min\{P_M(n), c_M\} - d, -B_{\max}\} \text{ のとき} \right. \\ \left. \left( C_{i, \min} \leq c_i \leq C_i, i=1, \dots, M, D_{\min} \leq d \leq D_{\max} \right) \right. \\ \left. 0, \quad \text{上記以外} \right. \end{array} \right. \quad (2.11)$$

さらに、状態  $s_n$  で決定  $\mathbf{a}$  をとったときの  $n$  期における直接費用は、

$$r(s_n, \mathbf{a}) = \sum_{i=1}^M \{ C_i^I I_i(n) + C_i^J [J_i(n)]^+ \\ + C_i^B [-J_i(n)]^+ + B_i H(J_i(n) < 0) \} \\ + C_{\max} \sum_{c_M = C_{M, \min}}^{C_M} \sum_{d = D_{\min}}^{D_{\max}} \Pr\{C_M(n) = c_M, D(n) = d\} \times \\ [d - B_{\max} - J_M(n) - \min\{P_M(n), c_M\}]^+ \quad (2.12)$$

で与えられる。ここで  $H(e)$  は、事象  $e$  が起これば値 1 を、起こらなければ値 0 をとる定義関数である。

この最適制御問題は、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でき、 $g$  を 1 期当たりの最小平均費用、 $h(s_n)$  を相対費用とおけば、次の最適性方程式が成り立つ。

$$g + h(s_n) = \min_{\mathbf{a} \in K(s_n)} \left\{ r(s_n, \mathbf{a}) + \sum_{s_{n+1} \in \mathbf{S}} p(s_n, s_{n+1}, \mathbf{a}) h(s_{n+1}) \right\}, \\ s_n \in \mathbf{S} \quad (2.13)$$

最適政策  $f^*(s_n)$  は、各  $s_n$  で(2.13)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。

### 3. プル方式とその最適設定アルゴリズム

プル方式としては、かんばん方式、基点在庫方式、CONWIP、ハイブリッド方式、拡張かんばん方式等が提案されているが、1. で述べたように拡張かんばん方式が最つとも優れた性能を示している。

拡張かんばん方式は、かんばん方式と基点在庫方式を組み合わせた方式であり、各工程  $i$  での発注量  $O_i(n)$  と生産指示量  $P_i(n)$  は次式で与えられる。

$$O_i(n) = \min \left\{ \begin{array}{l} D(n-1) + B_i(n-1), M_i - I_i(n) - [J_{i-1}(n)]^+ \\ - \sum_{l=1}^{L_i-1} O_i(n-l) - \sum_{l=0}^{T-1} Q_i(n-l) \end{array} \right\} \quad (3.1)$$

$$P_i(n) = \min \{ N_i - [J_i(n)]^+, I_i(n), C_i \} \quad (3.2)$$

ここで  $M_i$  は引取かんばん枚数、 $N_i$  は生産指示かんばん枚数である。 $B_i(n)$  は工程  $i$  における第  $(n-1)$  期までの最終製品の受注残であり、

$$B_i(n) = B_i(n-1) + D(n-1) - O_i(n) \quad (3.3)$$

で与えられる。しかし、この受注残は JIT サプライチェーンには存在しない状態変数であり、最適制御の初期政策として利用するためには、性能は落ちるものの、かんばん方式の方が適している。かんばん方式の発注量  $O_i(n)$  は周知のように、式(3.1)の第 2 項で与えられ、生産指示量  $P_i(n)$  は式(3.2)で与えられる。[3]ではプル方式の各パラメータを、平均総費用を最小化する値に設定するアルゴリズム SBOS (Simulation-based optimal setting) を提案している。また、[4]では生産能力が一定の単一工程 JIT 生産システムにたいする安定条件を導いているが、この条件よりかんばん枚数は、

$$M_i > (L_i + 1)D, N_i > D, i = 1, \dots, M \quad (3.4)$$

を満たさなければならない。以下、かんばん方式のパラメータ  $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_M)$  を最適に設定するアルゴリズム SBOS を述べる。ここで、 $\mathbf{X}_i = (M_i, N_i)$   $i = 1, \dots, M$  である。

遺伝的アルゴリズムやタブー探索法等を用いたプル方式の最適設定アルゴリズムは既に幾つか提案されている。しかしそれらは、全てのパラメータを一度に扱うため、多工程システムにたいしては、初期解による計算時間の変動

が大きく、不安定である。SBOS は、最終の単一工程から始めて工程数を 1 つずつ増やしなが、列挙法とタブー探索法を併用してパラメータを設定するアルゴリズムである。

#### SBOS (Simulation-based optimal setting)

1. 最終工程  $M$  のみのシステムにおいて、列挙法を用いてシミュレーションにより平均費用が最小になるパラメータ  $\mathbf{X}_M$  を求める。  $m = M-1$  とおく。

2. 上流へ 1 工程拡張した工程  $m \sim M$  のシステムを考え、工程  $(m+1) \sim M$  については求められているパラメータ  $(\mathbf{X}_{m+1}^*, \dots, \mathbf{X}_M^*)$  で固定し、工程  $m$  についてのみ列挙法を用いて平均費用が最小になるパラメータ  $\mathbf{X}_m^*$  を求める。

3. ステップ 2 で求められた  $(\mathbf{X}_m^*, \dots, \mathbf{X}_M^*)$  を初期解として、タブー探索法を用いて工程  $m \sim M$  のシステムの平均費用を最小化するパラメータ  $(\mathbf{X}_m^*, \dots, \mathbf{X}_M^*)$  を求める。ここで、近傍は現在のパラメータから増減 1 の範囲であり、タブーリストは、最近の  $L_T$  個の設定である。

4.  $m=1$  であれば終了。求められたパラメータ  $(\mathbf{X}_1^*, \mathbf{X}_2^*, \dots, \mathbf{X}_M^*)$  が解である。さもなければ  $(\mathbf{X}_m^*, \dots, \mathbf{X}_M^*) = (\mathbf{X}_{m-1}^*, \dots, \mathbf{X}_M^*)$  とし、  $m = m-1$  とおいてステップ 2 へ。

アルゴリズム SBOS では、与えられたパラメータに従い発注量と生産指示量を計算し、メルセンヌ・ツイスタ[5]を用いて離散分布から生産能力と需要量を乱数として生成している。そして、各状態から式(2.12)を用いて各期の費用  $r(n)$  を計算し、システムが定常に達するまでの期間を  $n_0$  として、  $n_0+1$  期以降の記録をとり、平均費用を推定している。SBOS では、平均費用の信頼区間を求めるために、バッチ平均法[例えば 6]を用いている。所定の精度の平均費用の推定値を得るためのアルゴリズムは以下のとおりである。

#### バッチ平均法

1. バッチ数  $f$  と望ましい信頼区間の幅  $\delta$  を定める。システムが定常に達するまでの期間  $n_0$  を定め、初期バッチ長  $b_0$  を定めて  $k=0$  とおく。

2. シミュレーション長  $b_j f$  のシミュレーションを実行し、バッチ平均  $r_j$ ,  $j = 1, \dots, f$  を求める。

3.  $r_1 < r_2 < \dots < r_f$  の場合は発散とみなしてシミュレーションを終了し、  $\bar{r} = \infty$  を出力する。

4. バッチ平均  $r_j$ ,  $j = 1, \dots, f$  からその平均  $\bar{r}$  と分散  $V^2$  を計算する。もし

$$t_{f-1,1-\alpha/2}V/\sqrt{f} \geq \delta \quad (3.5)$$

ならば  $b_{k+1} = 2b_k, k=k+1$  とおき,  $b_k < b_{max}$  ならばステップ 2 へ. ここで,  $t_{f-1,1-\alpha/2}$  は自由度  $f-1$  の  $t$  分布の  $1-\alpha/2$  点である.

5. 式(3.5) が成り立たなければ, 平均費用の  $1-\alpha$  信頼区間は

$$\left[ \bar{r} - t_{f-1,1-\alpha/2}V/\sqrt{f}, \bar{r} + t_{f-1,1-\alpha/2}V/\sqrt{f} \right] \quad (3.6)$$

である.

#### 4. 提案アルゴリズム

最適化されたかんぱん方式をシミュレーションし, SBMPIM に必要な平均費用と相対値を計算する. これらの値及び政策を初期値として使用する. SBMPIM の収束を加速できるものと期待される. 提案アルゴリズムは以下のとおりである.

1. (最適パラメータ設定) かんぱん方式のパラメータ  $M_i, N_i, i=1, \dots, M$  を SBOS により最適化する.

2. (初期設定) 初期状態  $s_0$  を最適かんぱんパラメータから設定し, シミュレーション長  $m$ , 相対値推定用加算数  $N$ , 停止基準  $\varepsilon > 0$  を定める. 相対値を計算した状態の集合  $S_Q = \phi$ ,  $S_T = \{s_0\}$ , 相対値を計算中の大きさ  $N$  の状態キュー  $S_H = \{s_0\}$ ,  $H(s_0) = 0$ , 累積費用  $TC = 0$ ,  $s = s_0$ ,  $l = 0$  とおく.

3. (Schweitzer 変換) 次式を満たす正数  $\tau$

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

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

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

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

#### 4. (かんぱん方式のシミュレーション)

4-1: 状態  $s$  で最適かんぱん政策  $f(s)$  をとったときの状態推移をシミュレーションし, 次期状態  $s'$  を定める.

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

$s_h \in S_H$  に対して  $H(s_h) = H(s_h) + r(s, f(s))$  を計算し,  $l < N-1$  ならば  $s = s'$ ,  $S_H = S_H + \{s'\}$ ,  $H(s') = 0$  と更新しステップ 4-2 へ. さもなければ

$$S_H = S_H - \{s_{n-N+1}\} + \{s'\}, \quad H(s') = 0$$

と更新する.

$$s_{n-N+1} \notin S_Q \text{ ならば } q(s_{n-N+1}) = 1, S_Q = S_Q + \{s_{n-N+1}\}, h(s_{n-N+1}) = H(s_{n-N+1})/N$$

$$s_{n-N+1} \in S_Q \text{ ならば } q(s_{n-N+1}) = q(s_{n-N+1}) + 1$$

$$h(s_{n-N+1}) = \{q(s_{n-N+1})h(s_{n-N+1}) + H(s_{n-N+1})/N\} / \{q(s_{n-N+1}) + 1\}$$

を計算し,  $s = s'$  と更新する.

4-2:  $s \notin S_T$  かつ  $l \leq m - N$  ならば  $S_T = S_T + \{s\}$ , 訪問回数  $v(s) = 1$  とおき,  $s \in S_T$  かつ  $l \leq m - N$  ならば  $v(s) = v(s) + 1$  と更新する.

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

#### 5. (平均費用 $g$ , 相対値 $h(s)$ の推定)

平均費用  $g$  を次式

$$g = TC/m\tau$$

により計算する.  $S_Q$  の中で  $v(s)$  が最大の  $s$  を  $s_r$  とおき,  $s \in S_Q$  に対して

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

を計算し  $h(s_r) = 0$  とおく.

#### 6. (収束判定)

$s \in S_Q$  に対して

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

を計算する. ここで  $p(s, s', f(s)) > 0$  となる  $s' \notin S_Q$  に対しては,

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

として  $w(s)$  を計算する.  $\{s \in S_Q \mid v(s) \geq 0.5v(s_r)\}$  に対して

$$\Delta(s) = |w(s) - g - h(s)|$$

を計算し,  $\max \Delta(s) > \varepsilon$  ならば  $s \in S_Q$  に対して

$$h(s) = w(s)$$

を計算してステップ 4 へ. さもなければステップ 7 へ.

#### 7. (初期設定 2)

$S_V = S_Q$  とおき,  $S_V$  の中で  $q(s)$  が最大の  $s$  を  $s_0$  とおき  $s \in S_V$  に対して次式を計算し相対値の精度を向上させる.

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

非負数  $\lambda$  ( $\lambda < 1$ ) 及び停止基準の正整数  $Q$ ,  $R$  及び  $\varepsilon', \varepsilon'' > 0$  を定める.  $S_U = \phi$ ,  $S_T = \{s_0\}$ ,  $TC = 0$ ,  $s = s_0$ ,  $k = l = 1$ ,  $q = 0$ ,  $g(0) = g$ ,  $W = \text{big}M$  とおく.

#### 8. (政策改良ルーチン)

8-1:  $s \in S_V$  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  $K(s)$  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,  $f(s')$  を最適かんぱん政策をとる決定と定め,

$$w(s') = r(s', f(s'))$$

$$h(s') = W$$

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.

8-2:  $s \in S_U$  に対して

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

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  $s' \notin S_V \cup S_U$  に対しては,  $h(s') = W$  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.

9. (シミュレーション)

9-1: 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,  $TC = TC + r(s, f(s))$ ,  $s = s'$

と更新する.

9-2:  $s \notin S_V$  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,  $f(s)$  を最適かんばん政策をとる決定と定め,  $w(s) = r(s, f(s))$  とおく.

9-3:  $s \notin S_V$  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.

9-4:  $s \in S_V$  かつ  $s \notin S_T$  のとき,  $S_T = S_T + \{s\}$ ,  $v(s) = 1$  とおく.

9-5:  $s \in S_V$  かつ  $s \in S_T$  のとき,  $v(s) = v(s) + 1$  と更新する.

9-6:  $l = m$  ならばステップ 10 へ. さもなければ  $l = l + 1$  としてステップ 9-1 へ.

10. (g の推定)

$S_T$  の中で  $v(s)$  が最大の  $s$  を  $s_r$  とおき, 1 期当たりの平均費用  $g(k)$  及び  $g$  を次式で計算する.

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

11. (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.

11-1:  $s (\neq s_0) \in S_T$  に対して,

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

とおく.

11-2:  $s \in (S_V \cup S_U) - S_T$  に対して,

$$h(s) = w(s) - g - h(s_0) \quad \text{とおく.}$$

11-3:  $h(s_0) = 0$  とおく.

12. (政策改良ルーチン)

ステップ 8 と同様

13. (収束判定)

13-1:  $|g(k) - g(k-1)| < \epsilon^n$  ならば,  $q = q + 1$  とおき, ステップ 13-2 へ. さもなければ  $q = 0$  とおき, ステップ 13-4 へ.

13-2:  $q = R$  ならば  $s_0 = s_r$  とおき, ステップ 13-4 へ. さもなければ  $q < Q$  のとき, ステップ 13-4 へ行き,  $q \geq Q$  ならば, ステップ 13-3 へ.

13-3:  $\{g(k-q), \dots, g(k)\}$  の標本分散  $S^2$  を平均  $g$  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,  $t_\alpha(q) S / \sqrt{q+1} < \epsilon'$  を満たせば停止. 最小平均費用  $g$  の  $100(1-\alpha)\%$  信頼区間は

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

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 13-4 へ.

13-4:  $s = s_0$ ,  $S_T = \{s_0\}$ ,  $TC = 0$ ,  $l = 1$ ,  $k = k + 1$  とおきステップ 9 へ.

5. 数値比較

[3] では提案アルゴリズムの基となる SBMPIM を 4200 万状態変数をもつ 3 工程サプライチェーンへ適用し, その準最適な最小平均費用を基準に最適化された各プル方式を比較している. まずその結果を紹介し, その後で提案アルゴリズムと従来の SBMPIM による計算結果を比較する.

図 1 の M 工程サプライチェーンにおいて,  $M=3, L_1=L_2=1, L_3=2, T_3=1, I_{max,1}=I_{max,2}=6, I_{max,3}=12, J_{max,1}=9, J_{max,2}=7, J_{max,3}=10$  とおき, 受注残の上限として  $B_{max}=5$  とおく. したがって, 状態数は 42,398,720 である. 費用係数として,  $(C_1^I, C_2^I, C_3^I) = (1, 3, 6)$ ,  $(C_1^J, C_2^J, C_3^J) = (3, 6, 12)$ ,  $(C_1^B, C_2^B, C_3^B) = (0, 0, 80)$ , 輸送中の在庫費用を  $C_3^Q = 6$ ,  $(B_1, B_2, B_3) = (0, 0, 120)$   $C_{max} = 1000$ . とおく. 需要分布は変形した二項分布

$$\Pr\left\{D_n = D - \frac{1}{2}Q + j\right\} = \binom{Q}{j} \left(\frac{1}{2}\right)^Q \quad 0 \leq j \leq Q$$

において平均  $D = 2$ ,  $Q = 2$  で与えられるものとし, 生産能力分布として次の 3 分布を考える.

- A:  $P_3=1$  (故障なし), 平均生産能力=3
- B:  $P_3=0.7, P_2=0.2, P_1=0.1$ , =2.6

C:  $P_3=0.7, P_1=0.2, P_0=0.1,$   $=2.3$

したがって、トラフィック密度は、A が 0.666, B が 0.769, C が 0.870 となり、C はかなり過密である。

表 1 は[3]の結果である。ここで、AAA は全工程の生産能力分布が A であり、CBA は第 1 工程の生産能力分布が C, 第 2 工程が B, 第 3 工程が A であることを示している。第 2 行目は従来の SBMPIM によって得られた準最適な最小平均費用の 95%信頼区間であるが、

VistaX64OS の Intel Core2 6700 2.66 GHz CPU, 8GB メモリーパソコンでの計算時間は、AAA で 6,272 秒, CCC で 303,369 秒であった。また、3 行目は最適かんぱん方式による平均費用の下限値と SBMPIM の上限値を比べた平均費用の増加率 (%) を次式

(最適かんぱん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) (%)

で求めた値を示しており、最後が SBOS による最適かんぱん枚数である。以下基点在庫方式, CONWIP, ハイブリッド方式, 拡張かんぱん方式にたいする同様な結果を示している。表 1 より拡張かんぱん方式が準最適政策に最も近いことが明らかとなった。

表 2 は同じ 3 工程サプライチェーンの最適制御問題へ提案アルゴリズムと SBMPIM を適用した最小平均費用を示している。使用した計算機は、Xeron3.33GHz CPU, 64GB ワークステーションである。共に、同様な結果が得られている。表 3 はそれらのパラメータ設定を示しており、表 4 は各々の計算時間を示している。表 4 より明らかに、提案アルゴリズムは従来の SBMPIM の計算時間を飛躍的に短縮することに成功している。さらに、表 1 と表 2 を比較することで、最適かんぱん枚数を利用した表 2 の SBMPIM は、表 1 の従来の SBMPIM では収束が困難であった問題 CCC, ABC, CBA に対して、より平均費用の低減された最適政策をより短時間で計算できることが示されている。

## 5. まとめ

本研究は、JIT 生産システムに基づく多品目多段階サプライチェーンの準最適な発注・生産・配送政策を計算できる、次元の呪いを克服したニューロ DP アルゴリズム SBMPIM の開発を目指すものである。そのため、最適化されたかんぱん方式をシミュレーションして得ら

れた平均費用および相対値を初期値とし、最適化されたかんぱん政策を初期政策とすることでより強力な SBMPIM を開発することができた。今後小売店、物流拠点を含むサプライチェーンへ適用する予定であるが、多品目サプライチェーンの準最適政策が計算できるためには、さらなる飛躍が求められる。

なお、本研究は科学研究費補助金・基盤研究(C)20510148 の補助を受けたものである。

## 参考文献

- [1] 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)
- [2] 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)
- [3] K. Ohno, “The Optimal Control of Just-in-time-based Production and Distribution Systems and Performance Comparisons with Optimized Pull Systems”, to appear in European Journal of Operational Research(2011).
- [4] K. Ohno, K. Nakashima and M. Kojima, “Optimal numbers of two kinds of kanbans in a JIT production system,” *Inter. Journal of Production Research*, Vol.33, No.5, pp.1387-1401(1995)
- [5] M. Matsumoto and T. Nishimura, “Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator,” *ACM Trans. on Modeling and Computer Simulation*, Vol.8, pp.3-30(1998)
- [6] 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)

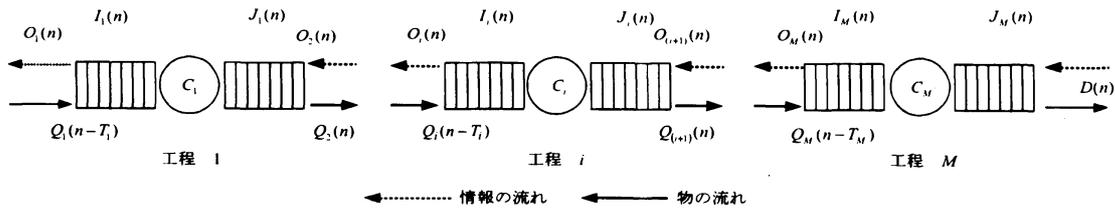


図1 M工程サプライチェーン

Production cap. dist.	AAA	BBB	CCC	ABC	CBA
SBMPIM	56.562±0.099	80.666±0.100	171.768±0.308	127.229±0.325	86.260±0.154
Kanban	69.859±0.350	92.199±0.966	210.236±1.261	158.678±0.964	92.044±0.836
	22.675%	12.960%	21.443%	23.645%	5.548%
	M:5,5,8	M:5,6,8	M:6,6,11	M:5,6,9	M:6,6,9
	N:3,3,3	N:3,3,4	N:9,8,10	N:3,4,9	N:7,3,3
Base stock	65.951±0.451	84.093±0.730	190.735±1.0139	138.856±0.997	98.214±0.899
	15.600%	3.215%	10.254%	8.079%	12.615%
	S:2,3,8	S:3,3,9	S:6,7,15	S:1,2,14	S:9,4,8
CONWIP	93.200±0.690	110.881±0.963	221.230±1.032	159.323±0.719	148.880±0.990
	63.269%	36.094%	27.966%	24.343%	71.141%
	S:17	S:18	S:30	S:21	S:22
Hybrid	81.097±0.318	104.602±0.825	211.910±1.037	179.632±0.732	91.534±0.748
	42.565%	28.491%	22.546%	40.254%	5.059%
	S:20	S:22	S:35	S:29	S:22
	M:5,8	M:5,9	M:6,9	M:6,10	M:6,9
Extended kanban	58.589±0.294	82.146±0.742	189.826±1.046	137.841±0.975	89.001±0.695
	2.884%	0.790%	9.707%	7.300%	2.189%
	M:6,6,9	M:6,9,10	M:9,11,22	M:5,6,12	M:6,7,9
	N:3,3,3	N:5,6,5	N:9,9,11	N:3,4,10	N:9,3,3
	R:0,0,3	R:0,0,0	R:0,0,11	R:0,0,6	R:0,0,3
S:1,2,7	S:3,2,10	S:6,0,11	S:0,1,10	S:9,2,7	

表1 3工程サプライチェーンにおけるSBMPIMによる準最適政策と最適化された各プル方式の比較[3]

Production cap. dist.	AAA	BBB	CCC	ABC	CBA
Proposal Algorithm	56.41±0.147	80.669±0.253	172.483±0.981	127.678±0.239	85.664±0.170
SBMPIM	56.494±0.017	80.634±0.972	172.436±0.954	127.926±0.992	85.716±0.764

表2 提案アルゴリズムとSBMPIMによる平均費用の比較

Production cap. dist.	AAA	BBB	CCC	ABC	CBA
Proposal Algorithm	$m=2.0 \times 10^4, N=10$ $\lambda=0.9, \tau=0.99, Q=20$ $R=1, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=1.0, \varepsilon''=1.0$	$m=5.0 \times 10^5, N=10$ $\lambda=0.9, \tau=0.99, Q=20$ $R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=1.0, \varepsilon''=3.0$	$m=5.0 \times 10^5, N=10$ $\lambda=0.9, \tau=0.99, Q=20$ $R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=1.0, \varepsilon''=3.0$	$m=5.0 \times 10^5, N=10$ $\lambda=0.9, \tau=0.99, Q=20$ $R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=1.0, \varepsilon''=3.0$	$m=5.0 \times 10^5, N=10$ $\lambda=0.9, \tau=0.99, Q=20$ $R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=1.0, \varepsilon''=1.0$
SBMPIM	$m=1.0 \times 10^5, \lambda=0.9, \tau=0.99$ $Q=20, R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=1.0$	$m=5.0 \times 10^5, \lambda=0.9, \tau=0.99$ $Q=20, R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=2.0$	$m=5.0 \times 10^5, \lambda=0.9, \tau=0.99$ $Q=20, R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=3.0$	$m=5.0 \times 10^5, \lambda=0.9, \tau=0.99$ $Q=20, R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=2.0$	$m=5.0 \times 10^5, \lambda=0.9, \tau=0.99$ $Q=20, R=3, \alpha=0.05$ $\varepsilon=1.0, \varepsilon'=2.0$

表3 提案アルゴリズムと SBMPIM のパラメータ設定

Production cap. dist.		AAA	BBB	CCC	ABC	CBA
Proposal Algorithm	Kanban	1	31	589	99	36
	SBMPIM	7	212	6518	449	92
	SBMPIM	1782	22074	266363	54907	19809

表4 提案アルゴリズムと SBMPIM による計算時間(sec)の比較