マルチエージェントシステムにおける学習エージェントの 数理モデル*

古川正志** 渡辺美知子** 大倉和博*** 嘉数侑昇#

A Study on Mathematical Modeling for Learning Multi-agents

Masashi FURUKAWA, Michiko WATANABE, Kazuhiro OHKURA and Yukinori KAKAZU

This paper presents a model for a multiple agent (multi-agent) system and a learning agent. The purpose of this study is to make it clear what is a reward for the learning agent and how the learning agent is constructed. The multi-agent system is defined as a pair of agents and their environment. A multi-agent system problem is defined as a set of the multi-agent system and its criteria so as to design the environment and agents maximizing the given criteria. An individual agent system is also defined as a set of the specified agent and his environment. Then, an individual agent problem is defined by a set of the specified agent, its environment, and its criteria. The agent reward is derived from this definition as an incremental value of the multi-agent system criteria. The agent learning process is formulated as a process of improving the agent criteria in order to maximize the agent criteria. Based on this analysis, necessary functions for the learning agent are derived. Namely, those are (1) a set of agent states sampled from the agent environment, (2) a set of agent behaviors, (3) the relational model between the agent state and behavior, (4) the estimating function for the criteria derived from the relational model, and (5) the agent behavior decision function. Finally, two cases are shown to construct the learning agent model using a gradient method and stochastic dynamic programming.

Key words: multi-agent system, multi-agent system problem, learning agent, adaptive learning, machine learning

1. 緒 言

近年、自律分散型生産システムの提案¹⁾や群ロボットなどの実現に向けてマルチエージェントシステム(multiple agent system)を基礎としたシステム構成が多く見られるようになってきている。一方、マルチエージェントシステムでは、エージェントに自律性、環境適応性を与えるため、その一手段として機械学習を採用する例が増加している。著者達も生産システムの搬送装置である無人搬送車(automatically guided vehicle: AGV)に対してこれをマルチエージェントとみなし、自律性、環境適応性を確保するために機械学習を採用した研究を実施している^{2)~4)}。

機械学習を採用する利点は、エージェントが与えられた環境に適応し、自律的な行動を行うための知識が未知の時に、そのような知識を試行錯誤によって獲得できる点に有る。しかし、こうした知識が明らかであれば機械学習を採用せずに自律性、環境適応性を有したエージェントを実現できる。従って、あえて機械学習を採用しなくともよい。

一方、機械学習に基づくエージェントの知識獲得は試行錯誤によって行われるため、エージェントに十分な経験を保有させるための時間が必要である。このことが機械学習に基づくエージェントの知識獲得の欠点ともなっている。そのため、知識獲得はその多くがオフラインで行われ、その後、オンラインで修正する方法が取られる。現実的には既知の知識を採用し、未知の知識に対し

てその獲得に機械学習を採用するのが良策である。

多くの研究に見られる機械学習を採用したエージェントの知識獲得問題では、エージェントをエージェントが認知する状態、認識する状態と行動の関係モデル、エージェントの行動に対する評価、エージェントの行動決定法、環境からの報酬等を唐突に定義し^{5),6)}、その後それらの具体化あるいは設計を行っている。このような唐突な定義は、環境とエージェントを一応モデル化しているとはいえ、それらの関係を具体的にモデル化し、エージェントの構成要件を導き出す研究に欠けるためと考えられる。本研究ではこうした観点から対象を人工のシステムに限定した上で、環境とエージェントをマルチエージェントシステムととらえ、そのモデル化を計ることによって、マルチエージェントシステムが本来的にもつ問題、環境からもたらされる報酬、エージェントが持つべき構成要件を明らかにすることを研究目的とする。マルチエージェントシステムを環境とエージェントのシステムとすることは、単にエージェントの集合をマルチエージェントシステムとする考え方とは異なる。

本論文の構成は以下である。2章でマルチエージェントシステムの定義とマルチエージェントシステムの問題の分類を行う。3章で環境からの報酬及びマルチエージェントシステム全体とエージェントの合目的性を導出する。更に、4章でエージェントの構成要件を明らかにし、5章ではエージェントの学習法を具体化する。最後に6章において結言を述べる。

2. マルチエージェントシステム問題

ここではマルチエージェントシステムを定義し、マルチエージェントシステムが本質的にもつ問題を明らかにする。

* 原稿受付 平成14年5月16日

** 正会員 旭川工業高等専門学校(旭川市春光台2-2)

*** 正会員 神戸大学(神戸市灘区六甲台町1-1)

正会員 北海道大学(札幌市北区北13西6)

Fig. 1 Multi-agent system

Fig. 2 Multi-agent problem

2.1 マルチエージェントシステム

吉川の一般過程設計論^{7),8)}では機械空間と機能空間の写像関係をインターアクションすることにより設計が進行する。本研究では人工的な環境とエージェントを機械空間にあてはめ、機能空間はそれに対応する環境とエージェントの機能と考え、機能空間にある環境とエージェントがどのように構成されているかを数理的に明らかにする。なお、ここで対象としている問題は人工物を対象とし、自然界に見られる環境とエージェントを取り扱っていない。また、環境とエージェントは人工物であり、明確に区別がつくとする。

環境とエージェントを以下のように定義する。

環境：エージェントを認識し、得られた情報にもとづいてエージェントの行動を制約する機能をもつ処理系

エージェント：環境を認識し、得られた情報にもとづいて自身の行動を決定する機能をもつ処理系

ここでは、環境及びエージェントの処理系を関数として表現し、それらを E 及び A と表す。ここで、

$$E \in \mathcal{E}, A \in \mathcal{A}$$

$\mathcal{E}$ 及び $\mathcal{A}$ は有限集合とし、それらの直積は先の機能空間を構成する。また、関数で表現した E 及び A を今後単に環境及びエージェントと呼ぶ。マルチエージェントシステム(MAS)を以下のように定義する(図1)。

$$MAS = (E, A) \quad (1)$$

すなわち、マルチエージェントシステムは環境とエージェントから構成されるシステムとする。いま、取り扱おうとする具体的なマルチエージェントシステムの対象を *Object* と記述すると、*Object* に対して上記表現をあてはめ、

$$\begin{aligned} \text{Object} &= (E, A) \\ \text{where } E &= \{E_1, E_2, \dots, E_\alpha\} \text{ and } A = \{A_1, A_2, \dots, A_\beta\} \end{aligned} \quad (2)$$

と表現する。環境は α 個の環境集合から構成され、またエージェントは β 個のエージェント集合から構成する。 E 及び A は情報処理を実施する関数の集合となり、環境 E はエージェント A が関数としてある情報処理を実施した結果を受け取り、また、エージェント A は環境 E が関数としてある情報処理を実施した結果を受け取る。このような *Object* に対しある評価を導入し、導入し

た評価を最適にするように環境 E とエージェント A を定める問題をマルチエージェントシステム問題 (Multiple Agent Problem: MAP) と定義する。すなわち、マルチエージェントシステム問題を

$$\begin{aligned} MAP &= (E, A, C) \\ \text{where } C &= \{C_1, C_2, \dots, C_\gamma\} \\ \text{and } C_k &: \mathcal{E} \times \mathcal{A} \rightarrow \mathcal{R}, (k=1, 2, \dots, \gamma) \end{aligned} \quad (3)$$

とする(図2)。式(3)において C_k は1次微分可能な写像関数、 x は直積集合演算、 $\mathcal{R}$ は実数空間を表わす。ここでは、システムの評価を一般的であるように γ 個で構成し、一般性を保つように多目的としてある。評価を最適にするように環境 E とエージェント A を定めるためには、MAP を上記3項組において具体的に以下の問題を解くことと定義する。

$$\text{Determine } E \text{ and } A \text{ such that maximize } C = C(E, A) \quad (4)$$

問題(4)において、 $\mathcal{E} \times \mathcal{A}$ は環境 E とエージェント A を決定する際の解空間となる。このような定式化は、マルチエージェント問題が以下のような3種類の設計問題となることを示している。

(1) 環境 E が既知の場合

$$\text{Determine } A \text{ such that maximize } C = C(E, A)$$

(2) エージェント A が既知の場合

$$\text{Determine } E \text{ such that maximize } C = C(E, A)$$

(3) 環境 E , エージェント A が共に未知の場合

$$\text{Determine } E \text{ and } A \text{ such that maximize } C = C(E, A)$$

具体的な例をあげれば、(1)は工場において工作機械群の配置とAGVの軌道が与えられた時に、AGVの行動を決定する問題、(2)はAGVの行動から工作機械群の配置とAGVの軌道を決定する問題、(3)はその両者を定める問題となる。太田らは最近になって環境とエージェントの共進化問題として(3)の具体的な問題を解いている⁹⁾。以下、本研究では(1)を取り扱いエージェントの構成要件を導くが、(2)に関しても同様の議論が可能である。

2.2 環境のダイナミックスの仮定

環境が時間の経過に従って変化する場合、環境を時間 t の関数として、

$$E = E(t) \quad (5)$$

と書くことができる。しかしながら、自然界や人工物の世界において環境の変化がダイナミックスであるような事例は少ないと考えられる。例えば、生物の進化において、環境を自然界、生物をエージェントと考えれば、自然界の変化はそこで生息する生物の変化に比してはるかに緩やかである。また、工場において機械群の配置を環境、そこでの作業者をエージェントと考えれば、新規の製造設備を工場に入れ替えたり、製造ラインを変えるまで、環境はほとんど変化しない。従って、ここで取り扱うMAPにおいては、環境はダイナミズムを持たないと仮定する。すなわち、

$$\left| \frac{dE(t)}{dt} \right| = 0 \quad (6)$$

を仮定し、取り扱う環境は静的であり、長時間変化しないとする。

3. 学習エージェントの報酬と構成要件

ここでは、エージェントが環境から学習を行う時の、エージェ

ントの報酬を明らかにし、またエージェントの構成要件を導出する。

3.1 個々のエージェントとその環境

いま、注目するエージェントを A_j とする。エージェント集合 A にとってその環境は E であるが、エージェント A_j にとっては、それ自身以外のエージェントも環境となる。いま、 A_j 以外のエージェント集合を $^{\wedge}A_j$ で表わすと、

$$^{\wedge}A_j = \{A_k : k = 1, 2, \dots, \beta, k \neq j\} \quad (7)$$

となる。 $^{\wedge}$ は否定集合を表わす。エージェント A_j の環境を E^*_j と表わせば、

$$E^*_j = (E, ^{\wedge}A_j) \quad (8)$$

とできる。これから、各エージェントを取り扱う対象としたとき、各エージェントから見たシステムを自身 A_j とその環境 E^* のシステム (agent system) $AS(j)$ において

$$AS(j) = (E^*_j, A_j)$$

と記述する。各エージェントもまたそれぞれ自身に与えられる評価を最適にするように行動するとすれば、個別のエージェントに対してそれぞれの以下の子問題 $ASP(j)$ を設定できる。

$$ASP(j) = (E^*_j, A_j, c_j) \text{ where } c_j = \{c_{j1}, c_{j2}, \dots, c_{j\delta}\}, \quad (9)$$

$$c_j = c_j(E^*_j, A_j) \text{ and } c_{jh} : E^*_j \times A_j \rightarrow \mathcal{R} (h = 1, 2, \dots, \delta)$$

ここで、各エージェントも評価は δ 個からなる多目的とする。各エージェントはそれ自身の評価に基づいて行動するとし、システムの評価とは別の評価をもつ。また、 c_{jh} は 1 次微分可能な写像関数とする。 $ASP(j)$ をエージェント A_j の具体的問題として、

$$\text{Given } E^*_j, \text{ determine } A_j \text{ such that} \quad (10)$$

$$\text{maximize } c_j = c_j(E^*_j, A_j)$$

と、具体化する。この子問題は、エージェント A_j がその周囲の状況が認識しながら、そのエージェントの機能を決定する問題となる。 E^* は他のエージェントを含むため動的となり、MAP で取り扱うような式(6)の仮定は成立しない。以下では、 E^* が与えられた場合を取り扱う。

3.2 マルチエージェントシステムとエージェントの評価の関係

式(4)におけるマルチエージェントの評価 $C = C(E, A)$ は、

$$C = C(E, A) \quad (11)$$

$$= C((E^*_1, A_1), (E^*_2, A_2), \dots, (E^*_\beta, A_\beta))$$

$$= C(AS(1), AS(2), \dots, AS(\beta))$$

$$= C(c_1(E^*_1, A_1), c_2(E^*_2, A_2), \dots, c_\beta(E^*_\beta, A_\beta))$$

と子問題の評価の関数とすることが可能である。上式はエージェント自身の評価とマルチエージェントシステム全体の評価の関係を表わす。すなわち、エージェント A_j が自身の評価を追求する時に、式(12)によってその評価は規制され、調節される。

今、エージェントが A_j から $A_j + \Delta A_j$ へと変化したとする。エージェントの変化は、エージェントの行動による離散的な値によって定まるが、ここでは連続的な変化として考察し、後の章で行動の導入に関して議論する。このとき、マルチエージェントシステムの評価の変化は、

$$\Delta C(A_j) = C(A_j + \Delta A_j) - C(A_j) \quad (13)$$

で書き表わされる。 $C(A_j + \Delta A_j)$ をテーラー展開して 1 次の項で近似すると

$$C(A_j + \Delta A_j) \approx C(A_j) + (\partial C(A_j) / \partial A_j) \Delta A_j \quad (14)$$

だから、

$$\Delta C(A_j) = (\partial C(A_j) / \partial A_j) \Delta A_j \quad (15)$$

を得る。従って、システムがエージェントの変化によって評価を改善するとすれば、

$$\Delta C(A_j) = (\partial C(A_j) / \partial A_j) \Delta A_j > 0 \quad (16)$$

である。一方、エージェント自身の評価の変化は、

$$\Delta c_j(A_j) = c_j(A_j + \Delta A_j) - c_j(A_j) \quad (17)$$

となる。従って、同様に

$$\Delta c_j(A_j) = (\partial c_j(A_j) / \partial A_j) \Delta A_j \quad (18)$$

を得る。また、

$$\partial C(A_j) / \partial A_j = [\partial C(A_j) / \partial c_j(A_j)] \cdot [\partial c_j(A_j) / \partial A_j] \quad (19)$$

から、

$$\Delta C(A_j) = [\partial C(A_j) / \partial c_j(A_j)] \cdot [\partial c_j(A_j) / \partial A_j] \cdot \Delta A_j \quad (20)$$

となり、これから $\partial c_j(A_j) / \partial A_j$ を求めると

$$\partial c_j(A_j) / \partial A_j = \Delta C(A_j) / [(\partial C(A_j) / \partial c_j) \cdot \Delta A_j] \quad (21)$$

である。従って、

$$\Delta c_j(A_j) = (\partial c_j(A_j) / \partial A_j) \Delta A_j$$

$$= [\Delta C(A_j) / \{(\partial C(A_j) / \partial c_j) \cdot \Delta A_j\}] \cdot \Delta A_j \quad (22)$$

$$= \Delta C(A_j) / (\partial C(A_j) / \partial c_j(A_j))$$

を得る。

3.3 エージェントの報酬と合目的性

式(22)において、 $r_j(A_j) = \Delta C(A_j)$ 、 $K = \partial C(A_j) / \partial c_j(A_j)$ とすれば、上式は

$$\Delta c_j(A_j) = r_j(A_j) / K \quad (23)$$

と書ける。 $r_j(A_j)$ はシステムを改善させる値であり、またエージェントの改善に与えられる値となる。これをエージェントの報酬と定義する。 ΔA_j の変化がシステムとエージェント自身の評価を同時に改善すると、 $\Delta c_j(A_j) > 0$ でなければならない。これから、

$$K = \partial C(A_j) / \partial c_j(A_j) > 0 \quad (24)$$

を得る。ここで、 $K > 0$ であれば、エージェントの評価が増加する時システムの評価が改善される。すなわち、エージェントの変化 ΔA_j によって、システムの評価とエージェントの評価は両者とも改善され、システムとエージェントは合目的となる。一方、 $K < 0$ であれば、エージェントの変化 ΔA_j はシステムの評価と相反し、非合目的となる。従って、エージェントの評価に関しては、 $K > 0$ となるようにエージェントを設計する必要がある。

3.4 学習

先に、エージェントの設計は、

$$\begin{aligned} &\text{Given } E^*_j, \text{ determine and } A_j \text{ such that} \\ &\text{maximize } c_j = c_j(E^*_j, A_j) \end{aligned} \quad (25)$$

とした。上記目的を果たすには、 A_j から $A_j + \Delta A_j$ に変化した時、 $r_j(A_j)$ かつ $K > 0$ で、 $\Delta c_j(A_j) > 0$ であればよい。従って、上記問題は、

$$\begin{aligned} &\text{Given } E^*_j, \text{ determine and } \Delta A_j \text{ such that} \\ &\text{maximize } c_j(A_j + \Delta A_j) = c_j(A_j) + r_j(A_j)/K \\ &\text{subject to } K > 0 \text{ and } r_j(A_j) > 0 \end{aligned} \quad (26)$$

と、 ΔA_j を決定する問題に変更できる。このとき、エージェントの評価は、

$$c_j(A_j + \Delta A_j) = c_j(A_j) + r_j(A_j)/K \quad (27)$$

で計算される。今、エージェントの変化によるシステム及びエージェント変化が、時間によるそれらの変化に比例する、すなわち

$$\frac{\Delta c_j(A_j)}{\Delta A_j} \propto \frac{\Delta c_j(A_j)}{\Delta t}$$

であると仮定する。エージェントが時刻毎に環境を認識し行動するとすると、この仮定は一般的に成立する。但し、 t は有限時刻をあらわす。このとき、問題(26)は、

$$\begin{aligned} &\text{Given } E^*_j, \text{ determine } \Delta A_j \text{ such that} \\ &\text{maximize } c_j(t + \Delta t) = c_j(t) + r_j(t)/K \\ &\text{subject to } K > 0 \text{ and } r_j(t) > 0 \end{aligned} \quad (28)$$

とでき、各時刻において上記目的を果たす ΔA_j を決定する問題となる。問題(28)を解く ΔA_j を決定するのが、学習過程である。

3.5 学習エージェントの構成

エージェントの変化は、エージェントがエージェントの備える行動（政策）によって導かれるとする。この時、エージェントの行動を

$$B_j = \{B_{j1}, B_{j2}, \dots, B_{jn}\}$$

と置く。ここで、

$$\Delta A_j = \pi_j(B_j), \text{ where } \pi_j: B_j \rightarrow \Delta A_j \quad (29)$$

とすれば、行動の選択がエージェントの変化となる。また、エージェントがエージェントの環境 E^*_j の一部しか認識できないとする。エージェントが認識した一部の環境を S_j と置くと、

$$S_j = Ob_j(E^*_j), \text{ where } S_j \subset E^*_j \text{ and } Ob_j: E^*_j \rightarrow S_j \quad (30)$$

とできる。エージェントがエージェントの環境 E^*_j の全てに出会う場合、

$$S_j = \{S_{j1}, S_{j2}, \dots, S_{j\xi}\}, \text{ where } \bigcup_{l=1}^{\xi} S_{jl} = E^*_j$$

とする。 $S_{j\xi}$ はエージェントが認識する環境であり、この手続きを環境のサンプリングと呼ぶ。サンプリングが十分な量の時エージェントはエージェントの環境 E^*_j 全体を把握できる。 S_j を、今後エージェントの状態と呼ぶ。これから問題(28)は、

$$\begin{aligned} &\text{Given } S_j, \text{ determine } B_j \text{ such that} \\ &\text{maximize } c_j(t + \Delta t) = c_j(t) + r_j(t)/K \\ &\text{subject to } K > 0 \text{ and } r_j(t) > 0 \end{aligned} \quad (31)$$

とできる。従って、エージェントは状態を入力とし、システムと評価が合目的であるように環境から報酬をもらい、内部に自身の評価計算をもち、それに基づいて行動の決定を行い、決定された行動を出力する。エージェントが自身の評価計算を実施するためには、状態から行動を計画するモデルが必要となる。評価計算はこのモデルに基づいて実施される。このモデルは、行動によって得られた報酬によって更新される。これらから、エージェントが必要とする構成要件は以下に示される。

$$A_j = (S_j, B_j, F_j, G_j, M_j, r_j) \quad (32)$$

各項目は、それぞれ

S_j : 状態

B_j : 行動

F_j : エージェントの評価関数 $c_j(t + \Delta t) = c_j(t) + r_j(t)/K$

G_j : 行動決定関数 $\text{maximize}_{B_j \in B_j} c_j(t + \Delta t) = c_j(t) + r_j(t)/K$

M_j : 環境 E^*_j の状態/行動モデル

となる。

4. エージェントの学習

ここでは、エージェントの学習を最大傾斜法とダイナミックプログラミングの二つの最適化の手法に基づいて具体化する。

4.1 最大傾斜法に基づく学習

よく知られた最大傾斜法を用いると、問題(25)または(26)は

$$\begin{aligned} A_j + \Delta A_j &= A_j + \lambda \partial c_j(E^*_j, A_j) / \partial A_j \\ &= A_j + \lambda r_j(A_j) / K \end{aligned} \quad (33)$$

あるいは、

$$A_j(t+1) = A_j(t) + \lambda r_j(A_j(t)) / K \quad (34)$$

が、最大化の過程となる。ここで、 $0 < \lambda < 1$ である。従って、エージェントの行動は、

$$\Delta A_j(t) = \lambda r(A_j(t)) / K / \Delta A_j \quad (35)$$

で決定される。エージェントの評価関数は

$$c_j(E^*_j, A_j(t+1)) = c_j(E^*_j, A_j(t)) + \lambda r_j(t) / K \quad (36)$$

となる。このような学習エージェントの例としては人工ニューラルネットワーク(ANN)のバックプロパゲーション法(BP法)がある。このとき、状態/行動モデル M_j は ANN のネットワークとなり、状態は入力層の入力、行動は出力層の出力、環境は教師信号、報酬はネットワークの教師信号と出力信号の誤差、行動決定関数が式(34)となりネットワークのウエイトを変更する。

4.2 確率的ダイナミックプログラミングによる学習 エージェント

$\Delta t = 1$ とし、 $t = 0, 1, 2, \dots$ と離散時間とする。問題(31)を各時刻で最適行動 B_j を得る最適化問題とすれば、問題(31)はダイナミックプログラミング(DP)の問題となる。今、行動 B_{jk} に対して、状態 S_{jk} が状態 S_{jv} に変わる確率が知られていれば、この確率を

$$P(B_{jk}) = [P_{uv}(B_{jk})]$$

と置き、これを状態遷移確率行列と呼ぶ。この状態遷移確率行列

をエージェントのモデルとする。また、時刻 t での行動を $B_{jk}(t)$ 、状態 S_j で行動 $B_{jk}(t)$ によって得られるエージェントの評価を $c_j(S_j, B_{jk}(t))$ と置く。離散時間 $t=0, 1, 2, \dots$ において、最適行動 $B_{jk}(t)$ を得るには、問題(31)から

$$\max_{B_{jk} \in B_j} c_j(S_{ju}, B_{jk}(t+1)) = r_j(B_{jk}(t))/K + \max_{B_{jk} \in B_j} c_j(S_{ju}, B_{jk}(t)) \quad (37)$$

であるように行動を B_{jk} が選択すればよい。一般に過去になるほど行動選択の影響は小さくするので、式(37)を、

$$\max_{B_{jk} \in B_j} c_j(S_{ju}, B_{jk}(t+1)) = r_j(B_{jk}(t))/K + \mu \max_{B_{jk} \in B_j} c_j(S_{ju}, B_{jk}(t)) \quad (38)$$

ととする。 μ は減衰率と呼ばれる。時刻 t での評価は、

$$c_j(S_{ju}, B_{jk}(t)) = \sum_v P_{uv}(B_{jk}(t)) c_j(S_{ju}, B_{jv}(t)) \quad (39)$$

と計算される。ここで、

$$c_j(B_{jk}(t)) = [c_j(S_{j1}, B_{jk}(t)) \quad c_j(S_{j2}, B_{jk}(t)) \quad \dots \quad c_j(S_{j\eta}, B_{jk}(t))]$$

$$r_j(B_{jk}(t)) = [r_j(S_{j1}, B_{jk}(t)) \quad r_j(S_{j2}, B_{jk}(t)) \quad \dots \quad r_j(S_{j\eta}, B_{jk}(t))]$$

なるベクトルを導入すると、式(37)は

$$\max_{B_{jk} \in B_j} c_j(B_{jk}(t+1)) = r_j(B_{jk}(t))/K + \mu \max_{B_{jk} \in B_j} P(B_{jk}(t)) c_j(B_{jk}(t)) \quad (40)$$

となる。各時刻で最適行動が得られ状態遷移にマルコフ過程が成立すれば、 t が十分大きい時、

$$c_j(B_{jk}(t)) = r_j(B_{jk}(t))/K + \mu P(B_{jk}(t)) c_j(B_{jk}(t)) \quad (41)$$

なる連立方程式を得る。これから、

$$c_j(B_{jk}(t)) = [I + \mu P(B_{jk}(t))]^{-1} r_j / K \quad (42)$$

を得る。各時刻で式(42)を解き、最大の評価をもつ行動を選択していくことにより確率的ダイナミックプログラミングは収束し、最適行動を導くことが証明されている¹⁰⁾。従って、確率的ダイナミックプログラミングによって、以下のように学習エージェントを構成できる。すなわち、モデルを状態遷移確率行列、エージェントの評価計算を式(42)、行動決定式を式(40)、状態を S_j 、行動を B_j とする。

なお、式(42)は

$$\begin{aligned} c_j(B_{jk}(t)) &= [I + \mu P(B_{jk}(t)) + (\mu P(B_{jk}(t)))^2 + \dots] r_j(B_{jk}(t))/K \\ &= \sum_{i=0}^{\infty} [\mu P(B_{jk}(t))]^i r_j(B_{jk}(t))/K \end{aligned} \quad (43)$$

と展開でき、時刻 t が十分大きい時に式(43)で最適行動が得られると、時刻 $t=0$ からの未来予測式となる。これを逐次式で表すと

$$c_j(B_{jk}(t+1)) = r_j / K + \mu P(B_{jk}(t)) r_j / K, \quad t = 0, 1, \dots \quad (44)$$

となる。式(44)の左辺を状態／行動評価の期待値、右辺の第1項を行動の報酬、第2項を状態／行動の推定評価値に置き換え、行

動選択に評価値のボルツマン分布による確率選択を採用すれば Q 学習の学習式となる。

5. 結 言

マルチエージェントシステム問題を環境とエージェントの観点からモデル化し、エージェントが学習機能を備えるときのエージェント構成を導いた。これらから以下を得られた。

- (1) マルチエージェントシステム問題は、エージェントと環境のどちらかあるいは双方を決定する問題と定義した。
- (2) エージェントが環境から受け取る報酬は、マルチエージェントシステムを改善する評価の増分である。
- (3) マルチエージェントシステムとエージェントの評価を一致させるには、パラメータ K を正值とする設計をする必要がある。
- (4) エージェントを設計するのに導かれたモデルから、具体的な学習エージェントを導入した。
- (5) エージェントが必要な構成機能は、サンプリング、報酬計算、モデル計算、エージェントの評価計算、行動決定計算、行動であることを示した。

本研究では、紙面に制約があるためエージェント間のコミュニケーション、エージェント間相互の学習、環境のモデル化に関する言及を行わなかった。これらについては改めて報告する予定である。なお、本研究は平成 13~15 年度科学研究費補助金基盤研究 C-2「学習・適応戦略をもつ生産システムエージェントの研究」に基づいて実施されたものであることを付記する。

参 考 文 献

- 1) T. N. Wang and C. K. Fan: APPLICATION OF AGENTS IN MANUFACTURING SYSTEM CONTROL, Proc. Of the 11th International Conference on Flexible Automation and Intelligent Manufacturing, Dublin City University, (2001) 835.
- 2) 渡辺美知子, 加藤龍, 古川正志, 木下正博, 嘉数侑昇: Q 学習による AGV 衝突回避通信プロトコルの獲得, 精密工学会誌, 66, 1 (2000) 107.
- 3) 渡辺美知子, 加藤龍, 古川正志, 木下正博, 嘉数侑昇: 簡易 S DM を利用した強化学習による AGV の経路獲得に関する研究, 日本機械学会論文集 C 編, 66, 643 (2000) 913.
- 4) 渡辺美知子, 古川正志, 木下正博, 嘉数侑昇: Q 学習による多数 AGV の自律搬送に関する研究, 精密工学会誌, 67, 10 (2001) 1609.
- 5) K. Narendra and M.A.L. Thathachar: LEARNING AUTOMATA AN INTRODUCTION, Prentice-Hall International (1989) 103.
- 6) J. H. Holland: ADAPTATION IN NATURAL AND ARTIFICIAL SYSTEMS, MIT Press (1992) 20.
- 7) 吉川弘之: 一般設計学序説——一般設計学のための公理的方法——, 精密機械, 43, 1 (1977) 21.
- 8) 吉川弘之: 一般設計過程, 精密機械, 45, 8 (1977) 19.
- 9) 坪井泰憲, 太田順, 千葉龍介, 新井民夫: 共進化手法を用いた AGV システムの統合的設計, 2002 年度精密工学会春季大会講演論文集, 精密工学会 (2002) 234.
- 10) G. Hadly: NONLINEAR AND DYNAMIC PROGRAMING, ADDISON-WESLEY PUB. CO. (1972) 454.