グラフ分割問題における多教師付き学習オートマトン計算法

正員 池坊 繁屋* 正員 銭 飛*
正員 平田 廣則**

A Multi-Teacher Learning Automata Computing Model for Graph Partitioning Problems

Shigeya Ikebo*, Member, Fei Qian*, Member, Hironori Hirata**, Member

Graph partitioning is an important problem that has extensive applications in many areas, including VLSI design, scientific computing, data mining, geographical information systems and job scheduling. The graph partitioning problem (GPP) is NP-complete. There are several heuristic algorithms developed finding a reasonably good resolution. The most famous partitioning methods are simulated annealing (SA) and mean field algorithm (MFA) known to produce good partition for a wide class of problems, and they are used quite extensively. However these methods are very expensive in time and very sensitive in parameters tuning methods.

In this paper, a new parameter-free algorithm for GPP has been proposed. The algorithm has been constructed using the S-model learning automata with multi-teacher random environments. As shown in our experiments, the proposed algorithm has some advantages superior to SA, MFA and ParMeTiS.

キーワード：グラフ分割問題，学習オートマトン，多教師未知環境

Keywords: graph partitioning problems, learning automata, multi-teacher environments

1. はじめに

本論文では，組合せ最適化問題の1つであるグラフ分割問題 (GPP: Graph Partitioning Problem) について考察する。GPP は，VLSI 設計問題，並列処理システムのジョブ割当て，スーパーバイプライン処理システムにおけるステージ内のプロセッサの自動割当て問題などにおいて主要な役割をする⁽¹⁾⁽²⁾。

ノード集合 $\{n_1, n_2, \dots, n_N\}$ ，リンク集合 $\{e_1, e_2, \dots, e_L\}$ のグラフ G が与えられた場合，GPP は次のような制約条件のもとで K 個の部分グラフ $\{G_1, G_2, \dots, G_K\}$ に分割する問題である (K-GPP)。

- 部分グラフ間のリンクコストの総和を最小にすること。
- 各部分グラフに属するノード数をなるべく指定された数にすること[†]。

- 1つのノードは1つの部分グラフにだけ属すること。

一般的に，均一分割問題の場合，これらの制約条件を満たすような評価関数は以下のように構築することができる。

$$f = \sum_{i=1}^N \sum_{j \neq i}^N \sum_{k=1}^K c_{ij}(1 - s_{jk})s_{ik} - \alpha \sum_{i=1}^N \sum_{j \neq i}^N \sum_{k=1}^K (1 - s_{jk})s_{ik} + \beta \sum_{i=1}^N \sum_{k=1}^K \sum_{m \neq k}^K s_{ik}s_{im} \quad \dots\dots\dots (1)$$

ここで，右辺の各項はそれぞれ上記の各制約条件と対応しており， $\alpha > 0, \beta > 0$ は各制約条件への依存性をあらわすパラメータである。また， c_{ij} はノード n_i, n_j 間のリンクコストであり， s_{ik} はノード n_i が部分グラフ G_k に割当てられているかどうかを示すためのものである。

$$s_{ik} = \begin{cases} 1, & n_i \in G_k, \\ 0, & n_i \notin G_k. \end{cases} \quad \dots\dots\dots (2)$$

[†]均一分割問題の場合，各部分グラフでノード数が同じであることが要求される。非均一分割問題の場合，事前に指定された部分グラフのノード分布が $\{N_1, N_2, \dots, N_K\}$ に満たされるように分割する必要がある。

* 広島国際学院大学
〒 739-0321 広島市安芸区中野 6-20-1
Hiroshima Kokusai Gakuin University
6-20-1, Nakano, Aki-ku, Hiroshima 739-0321

** 千葉大学
〒 263-8522 千葉市稲毛区弥生町 1-33
Chiba University
1-33, Yayoi-cho, Inage-ku, Chiba 263-8522

一方、非均一分割問題の評価関数は、式 (1) の第 2 項が以下のように修正される。

$$+\alpha \sum_{k=1}^K |N_k - \sum_{i=1}^N s_{ik}| \dots\dots\dots (3)$$

但し、 N_k ($k = 1, 2, \dots, K$) は部分グラフ G_k のノード数である。

大規模 GPP のための計算法は多数提案されている。理論的に成功している計算法としては、シミュレーテッドアニーリング法 (SA: Simulated Annealing Algorithm) ^{(3) (4)} や平均場近似計算法 (MFA: Mean Field Annealing) ^{(5) (6)} が挙げられる。

SA は物性の結晶構造を作る過程を模擬しており、学習過程において評価関数の値が悪くなるような解を確率的に与えることにより局所解からの脱出を可能にする確率的手法である。SA を用いて GPP を解く場合、問題の解空間の構造に関する情報が得られないために各温度でのサンプル様相を実現するためには十分に長い時間で観測しなければならない⁽⁷⁾。MFA は SA の計算過程に並列分散的計算手法を導入したものであり、スピングラス理論の Weiss 近似理論に基づき、隣接要素 (スピン) 間の相関関係を系全体の平均値で近似し平均値の計算を並列化することが可能となる。SA に比べ MFA は収束速度が速い利点を持っているが、現在までに提案されている並列化平均場近似計算法の適用例から考察すると Weiss 近似から生ずる解の精度の問題点が露呈する。これは、統計学の分野で一般に良く知られている⁽⁸⁾、相関関数の測度空間上での距離依存性、及び温度パラメータ依存性が無視されているので、系の性質表現が臨界指数近傍では正確でないことによる。

筆者らは、 N 個の確率オートマトン (StA: Stochastic Automaton) からなる学習オートマトン (LA: Learning Automaton) ^{(10) (11)} チームモデルを用いて、GPP のための単一教師付き LA 計算法を提案した⁽⁹⁾。GPP のための LA 計算法は SA に近い計算精度を保ちながら、MFA 並みの計算速度を持つことが特徴的である。

GPP において、現在までに提案されている計算手法の多くは式 (1) の評価関数、またはその変形を利用している。式 (1) からわかるように、各制約条件の強さを表す依存度パラメータの設定は解の制度に重要な影響を及ぼすことになるが、一般的には GPP の規模に応じて、実験上で試行錯誤的に決めることが多い。

本論文では、GPP のために多教師付き LA (MLA: Multi-teacher LA) を導入することにより、解が各制約条件の依存状況を動的に推定することにより、評価関数 (式 (1)) の各制約条件を全て最大限まで満たす計算法を提案する。そして、提案手法の有効性を示すために、既存の幾つかの代表的な計算法を用いて比較検証を行う。

2. 学習オートマトンモデル

LA は StA とそれを取り囲む未知環境 (RE: Random En-

図 1 多教師付き LA のチームモデル

Fig. 1. Multi-teacher learning automata model.

vironment) から構成される。LA を GPP に適用するために、グラフ G の各ノード上に StA を 1 つずつ配置させ、各部分グラフへの割当て状況を監視する。評価関数の各制約条件への満足度はそれぞれの未知環境で対応させる。これにより、各制約条件のトレードオフ関係を最大限に満たすよう、StA 群はチーム的動作を行う。この場合、未知環境はオートマトンの戦略決定機構への評価を与える意味で教師と呼ばれている。筆者らは先に提案した手法では、GPP の評価関数 (式 (1)) を 1 つの未知環境と見なし、探索過程における評価関数値の変動を教師信号として利用していた。その際に、各制約条件への依存性をあらわすパラメータ α と β を実験パラメータとして扱ってきた。

本論文では、探索結果の実験パラメータへの依存性を解消するために、トレードオフ関係にある式 (1) の 3 つの項を 3 つの未知環境で対応させ、GPP のための多教師付き LA 探索手法を提案する。

[定義 1]

$$LA = \{StA^i, RE^j\}, i = 1, \dots, N, j = 1, \dots, M \dots\dots\dots (4)$$

〈2・1〉 確率オートマトン

[定義 2]

$$StA^i = \{Y^i, r^i, \Phi^i, p^i, o^i, T^i\} \dots\dots\dots (5)$$

- Y^i : 出力集合, $Y^i = \{y_1^i, \dots, y_K^i\}$
- r^i : 強化信号, $r^i \in [0, 1]$
- Φ^i : 状態空間, $\Phi^i = \{\phi_1^i, \dots, \phi_K^i\}$
- p^i : 出力選択確率分布, $p^i = (p_1^i, \dots, p_K^i)^T$
- o^i : 出力関数^{††}, $o^i: \Phi^i \rightarrow Y^i$
- T^i : 強化法

^{††}通常では、出力関数は単射写像であると仮定されている。従って、本来の「確率分布に従い、状態遷移を決定し、出力関数を用いて出力を選択する」ということは、「確率分布に従って、出力を選択する」とことと同じ意味を持つ。

図1には GPP のための LA のチームモデルを示している。ここで、K-GPP について考える。

チームは N 個のオートマトン $\{StA^1, StA^2, \dots, StA^N\}$ からなり、各ノード上に置かれたオートマトン StA^i はそれぞれ K 個の出力を持ち、時刻 t において、 StA^i がノード n_i を k 番目の部分グラフ G_k に割当ててることを、出力 $y^i(t) = y_k^i$ で表す。各 StA の出力選択の方針は、ノードの割当てに伴う評価関数のコスト値が減少する方向に向かわせ、独立的に自分の戦略の強化及び決定を行うことである。各 StA にとってチーム行動中での目標とは、自身が未知環境から受ける平均損失を最小とする混合出力 $(y_{k1}^1, y_{k2}^2, \dots, y_{kN}^N)$ を見出すことである。明らかに、GPP のための LA のチームモデルはチーム型非協力ゲームに帰着され、ゲームのプレイは混合出力、ゲームの結果は各環境から受ける損失(プレイ後評価関数の変化値が減っているなら小さい損失が与えられる)である。このようなチーム型非協力ゲームには、ゲームの戦略空間上にナッシュ均衡点(最適混合戦略)が存在することは既に証明されている⁽¹²⁾。

評価関数(式(1))の最小値を推定するために、本論文では、未知環境からの損失値が連続値で表される S-model($r^i \in [0, 1]$) の LA を利用する。この場合、出力とそれに対する未知環境からの応答に応じた出力選択確率分布 p^i の修正は、以下のような強化法 T^i を用いて行う。

時刻 t での出力を $y^i(t) = y_k^i$ としたとき、出力選択確率分布 p^i は以下のように修正される⁽¹³⁾ †††。

$$p_k^i(t+1) = p_k^i(t) + \theta^i \sum_{j \neq k}^K [h(d_k^i(t) - d_j^i(t))] \cdot \left[\delta_{kj}(t) p_j^i(t) + \delta_{jk}(t) \frac{(1 - p_j^i(t)) p_k^i(t)}{K-1} \right] \quad (6)$$

$$p_j^i(t+1) = p_j^i(t) - \theta^i [h(d_k^i(t) - d_j^i(t))] \cdot \left[\delta_{kj}(t) p_j^i(t) + \delta_{jk}(t) \frac{(1 - p_j^i(t)) p_k^i(t)}{K-1} \right] \quad (j \neq k) \quad (7)$$

但し、

$$\delta_{kj}(t) = \begin{cases} 1 & \text{if } d_k^i(t) > d_j^i(t) \\ 0 & \text{if } d_k^i(t) \leq d_j^i(t) \end{cases} \quad (8)$$

$$\begin{cases} d_k^i(t+1) = \frac{R_k^i(t+1)}{z_k^i(t+1)} \\ R_k^i(t+1) = R_k^i(t) + r_k^i(t) \\ z_k^i(t+1) = z_k^i(t) + 1 \end{cases} \quad (9)$$

$$\begin{cases} d_j^i(t+1) = d_j^i(t) \\ R_j^i(t+1) = R_j^i(t) \\ z_j^i(t+1) = z_j^i(t) \end{cases} \quad (j \neq k) \quad (10)$$

†††文献中では P-model($r^i \in \{0, 1\}$) を用いているが、本論文では S-model として用いている。

ここで、 z_j^i は出力 y_j^i が選択された回数、 R_j^i は現時点までの r_j^i の総和である。明らかに、 d_j^i は出力 y_j^i の評価値である平均リワードである。また、 h は単調増加関数 $[-1, 1] \rightarrow [-1, 1]$ であり、 $0 \leq \theta^i \leq 1$ は学習パラメータである。

〈2・2〉 未知環境

〔定義 3〕

$$RE^j = \{Y^j, r^j, f^j\} \quad (11)$$

Y^j : 入力集合、 $Y^j = \{y^1, \dots, y^N\}$

r^j : 強化信号

f^j : 評価関数

未知環境を以下のようにモデル化する。

評価関数(式(1))の各項の推定値を以下の関数で表し、それぞれの未知環境に対応させる。

$$f^1(t) = \sum_{i=1}^N \sum_{j \neq i}^N \sum_{k=1}^K c_{ij}(1 - s_{jk}(t)) s_{ik}(t) \quad (12)$$

$$f^2(t) = \sum_{i=1}^N \sum_{j \neq i}^N \sum_{k=1}^K (1 - s_{jk}(t)) s_{ik}(t) \quad (13)$$

$$f^3(t) = \sum_{i=1}^N \sum_{k=1}^K \sum_{m \neq k}^K s_{ik}(t) s_{im}(t) \quad (14)$$

ここで、式(1)におけるパラメータ α, β が現時点において存在しないことに注意されたい。また、評価関数値を最小にするために、各未知環境からの強化信号 r^j は以下のような値を取るものとする。

$$r^1(t) = h(f^1(t)) \quad (15)$$

$$r^2(t) = g(f^2(t)) \quad (16)$$

$$r^3(t) = h(f^3(t)) \quad (17)$$

但し、

$$g(x) = \frac{1}{1 + e^{-x}} \quad (18)$$

$$h(x) = \frac{e^{-x}}{1 + e^{-x}} \quad (19)$$

チーム的ゲームモデルで複数 (M 個) の未知環境を考える場合、各未知環境からの強化信号の集合 $r(t) = (r^1(t), r^2(t), \dots, r^M(t))^T$ を便利上実数空間 $[0, 1]$ に写す正規化写像 $G: r \rightarrow [0, 1]$ を定義する必要がある。まず考えられる方法としては、以下のような重み付き多数決法が考えられる。

(1) 重み付き多数決法

$$r(t) \equiv G(r(t)) = \frac{1}{M} w^T(t) \cdot r(t) \quad (20)$$

但し,

$$\mathbf{w}(t) = (w^1(t), \dots, w^M(t))^T, \quad w^j(t) \in [0, 1]$$

ここで, 明らかに $\mathbf{w}$ は各制約条件の依存度を表すパラメータであり, 重み付き多数決法で学習を行う場合, 計算法自体が従来の LA 計算法に帰着され, 評価関数のパラメータの試行錯誤的調節が避けられない。そこで, 本論文では以下のような累積誤差法を用いる。

(2) 累積誤差法

$$\begin{aligned} r(t) &\equiv G(\mathbf{r}(t)) \\ &= \frac{\left[\tilde{r}^i(t) - \min_j \tilde{r}^j(t-1) \right]^+}{\left[\max_k \left(\tilde{r}^k(t) - \min_j \tilde{r}^j(t-1) \right) \right]^+ + 1} \end{aligned} \quad (21)$$

但し,

$$\tilde{r}^i(t) \equiv \frac{\sum_{n=t-l}^t r^i(n) \eta(y^i(n) = y^i)}{\sum_{n=t-l}^t \eta(y^i(n) = y^i)} \quad (22)$$

$$[x]^+ = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

$$\eta(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

ここで, $1 < l < t$ は観測空間の長さを示しているパラメータであり, 全空間について考える場合, 上記正規化関数は Poznyak の正規化関数に一致している。観測空間のサイズ l を大きく取りすぎると学習過程に惰性による過学習反応が現れ, 必ずしも良いとは限らない。本論文での実験検証では $l = 2$ としている。

3. アルゴリズム

以下に GPP における多教師付き LA 計算法 (MLA) のアルゴリズムを示す。

Initialize

- ノード集合をランダムに部分グラフに割当てて。
- 各ノード上に StA を 1 つずつ配置させる。
- 各 StA^i の出力選択確率分布 $\mathbf{p}^i$ を初期化する。

$$\mathbf{p}^i(0) = (1/K, \dots, 1/K)^T, \quad i = 1, \dots, N.$$

Method

Step 1:

時刻 t において, ランダムに 1 つのノード n_i (StA^i) を選択する。

Step 2:

選択された StA^i において, 出力選択確率分布 $\mathbf{p}^i$ に従い出力 y_k^i (割当て部分グラフ) を決定し, その部分グラフに割当てて。

Step 3:

各未知環境において各自の評価関数を用いて計算をおこない, 累積誤差法を用いて 1 つの強化信号 $r(t)$ を StA^i に返す。

Step 4:

StA^i において, 式 (6), (7) の強化法を用いて出力選択確率分布 $\mathbf{p}^i$ の修正を行う。

Step 5:

終了条件を満たすまで Step 1 に戻る。

この動作を繰り返すことにより最小損失を受け取る最適出力を決めていく。この最適出力は GPP の最適割当てに相当する。

4. 計算機実験

提案している手法を検証するために, 表 1 に示す実験用グラフセットについて分割数 $K=10, 50, 100$ として計算機実験を行った。

表 1 中の U_N はノード数が N , $4Nd = 192.08$ である geometric グラフ (図 2-a) であり, d はクラスタリング半径である。 M_N はノード数が N である 2 次元メッシュグラフ (図 2-b) であり, リンク数は $2(\sqrt{N}-1)\sqrt{N}$ で算出される。 T_N はノード数が N である 2 次元トライアングルメッシュグラフ (図 2-c) であり, リンク数は $3(\sqrt{N}-1)\sqrt{N}-1$ で算出される。

本提案手法と比較対象である各手法の実験パラメータは

表 1 実験用グラフセット。

Table 1. Experimental graph sets.

Graph Name	No. of Nodes	No. of Links	Graph Type
U600	600	3513	2D random graph
U1000	1000	5982	2D random graph
U1900	1900	10152	2D random graph
U5000	5000	36987	2D random graph
M400	400	760	2D mesh
M10K	10000	19800	2D mesh
M90K	90000	179400	2D mesh
T40K	40000	198005	2D triangle mesh
T62K	62500	186501	2D triangle mesh

図 2 実験検証用データセット

Fig. 2. Experimental graphs.

以下の通りである。ここでの各パラメータの値は経験値であり、学習パラメータ θ^i は式 (6),(7) の強化法で用いるパラメータである。

シミュレーテッドアニーリング法 (SA) :

初期温度: 2000, 終了温度: 0.001, 初期割当て方法: ランダム法, 各温度での反復回数: 4000, 定数降温法の降温係数: 0.95

平均場近似計算法 (MFA) :

初期温度: 100, 終了温度: 0.001, 初期割当て方法: ランダム法, 各温度での反復回数: 1000, 定数降温法の降温係数: 0.97

固定構造学習オートマトン計算法 (FLA) ⁽⁹⁾ :

学習パラメータ $\theta^i = 0.001$, 初期割当て方法: ランダム法, 終了条件: コストの変動値が連続 100 回 0.0001 以内。

多教師付き学習オートマトン計算法 (MLA) :

学習パラメータ $\theta^i = 0.0037$, 初期割当て方法: ランダム法, 終了条件: コストの変動値が連続 100 回 0.0001 以内。

なお、シミュレーションの実行時間を評価するために、ここでは有名な並列化手法 **ParMeTiS** と比較する。ParMeTiS は PC-Cluster システム上で MPI(Message Passing Interface) により実装された GPP ソフトウェアであり、大規模 GPP における計算速度の速さと高い精度で有名である⁽¹⁴⁾。ここでは ParMeTiS をデフォルトパラメータ設定状態で、32 ノードの PC-Cluster システム上で実行している。本論文で提案している手法も MPI で並列化を行い、ParMeTiS と同じ計算環境上で実行している。他の手法については、同じ PC-Cluster システム上の単一ノードを使って計算している。

実験結果を図 3~9 に示す。但し、各手法における実験結果の数値は、SA, MFA, FLA は試行回数 10 回の平均値であり、ParMeTiS, MLA は試行回数 20 回の平均値である。

図 3, 5, 7 は、各グラフに対するコスト結果を分割数ごとにまとめている。ここでのコストとは、各手法とも最終的に

図 3 10 分割問題のコスト比較

Fig. 3. Costs comparison of 10-way partitions.

図 4 10 分割問題の計算時間比較

Fig. 4. Convergence time comparison of 10-way partitions.

図 5 50 分割問題のコスト比較

Fig. 5. Costs comparison of 50-way partitions.

図 6 50 分割問題の計算時間比較

Fig. 6. Convergence time comparison of 50-way partitions.

各ノードがどの部分グラフに属するか明確であることから、式 (1) の第 1 項により得られる値である。すなわち、最終的に得られる部分グラフ間のリンクコストの総和を示している。また、図 4, 6, 8 は、そのときの計算時間を示している。なお、SA における M90K(K=50,100), T62K(K=100) については、計算時間の都合により結果を得ることが出来な

図7 100分割問題のコスト比較

Fig. 7. Costs comparison of 100-way partitions.

図9 ParMeTiSに対する提案手法の精度誤差

Fig. 9. Costs deviations from MLA to ParMeTiS.

図8 100分割問題の計算時間比較

Fig. 8. Convergence time comparison of 100-way partitions.

かった。

図3,5,7におけるコストの比較結果より、いずれの分割問題においても提案手法はSAおよび他の手法並みの精度を有していることが分かる。例えば、グラフM400に対する各分割問題(K=10,50,100)のコスト結果は、全ての手法において最適値(9.0, 49.0, 99.0)が得られている。また、図4,6,8における計算時間の比較結果より、提案手法はParMeTiSなみの高速性を有していることは明らかである。また、SA, MFA, FLA法においては降温係数や α, β などの実験パラメータの事前調節時間が必要であることを考えれば、実験パラメータの調節が不要である提案手法の有効性がさらに明らかである。

ここで、ParMeTiSとのコスト差をより顕著に見るために提案手法との相対的な精度誤差(0軸以下のとき、ParMeTiSより改善していることを示す。)を比較した結果を図9に示す。提案手法は、各グラフや分割数によってはParMeTiSに比べ高いコスト結果を得ているが、全体的に見ると、提案手法の方がより低いコスト結果を得ていることが多いことより、提案手法が精度において若干優越であることが分かる。

なお、紙面の都合で示すことができないが、非均一分割

問題についても同等な計算結果が実験で得られている。

5. おわりに

本論文では、グラフ分割問題に対する多教師付き学習オートマトン計算法を提案している。

GPPの評価関数における各制約条件を複数の未知環境に対応させ累積誤差法を用いることにより、提案手法は学習アルゴリズムにおける実験パラメータ α, β の事前調節が不要である。従来までは、これらのパラメータは各制約条件への依存度を示しているが、本論文で提案している手法では、LAを導入することにより全ての制約条件を同時に最大限まで満たすことができる。これは、式(21)を利用したLAチームが過去一定期間における最小平均損失 $\min_j \bar{\epsilon}_j$ への相対的な精度誤差を各自自動的に調整することにより達成されと考えられる。提案手法はLAのチームモデルを基本にしてモデル化を行っているため、非協力的ゲームの特性から並列化しやすい利点を持っている。

また、提案手法中ではS-modelの強化法を採用することにより、評価関数値の評価を高精度で行うことができるようになっている。

提案手法の有効性を示すために、本論文では9セットのグラフデータを用いて他の手法との比較検証を行った。実験の結果で示されたように、限定時間内において提案手法は良い精度と良好な計算速度を持つことが分かった。

(平成14年5月20日受付, 平成14年10月21日再受付)

文 献

- (1) <http://rtm.science.unitn.it/intertools/graph-partitioning/links.html>: Graph Partitioning - WWW Resources.
- (2) 横山隆志・池坊繁屋・銭 飛:「大規模組合せ最適化問題のための疎結合並列計算機システム」, 電気学会電子情報システム部門全大論文集 II, pp.665-670 (2001)
- (3) C.D.Gelatt Jr., S.Kirkpatrick, and M.P.Vecchi: "Optimization by simulated annealing", *Science*, Vol.200, pp.671-680 (1983)
- (4) Emile Arts and Jan Korst: *Simulated Annealing and Boltzmann Machines*, John Wiley & Sone (1990)
- (5) D.E.Van den Bout and T.K.Miller, III: "Graph partitioning

using annealed neural networks", *IEEE Trans. Neural Net.*, Vol.1, No.2, pp.192-203 (1990)

- (6) C.Peterson and E.Hartman: "Explorations of mean field theory learning algorithm", *Neural Networks*, Vol.2, pp.475-494 (1989)
- (7) 深尾 毅: 分散システム論, 昭晃堂 (1987)
- (8) M.Mezard, G.Parisi, and M.A.Vieasoro: "Spin Glass Theory and Beyond", *World Scientific Publishing Co. Pte. Ltd* (1987)
- (9) 銭 飛・池坊繁屋・平田廣則:「グラフ分割問題のための学習オートマトン計算法」, 電学論誌 C, **120**, 3, pp.365-371 (2000-3)
- (10) K.S. Narendera and M.A.L. Thathachar: "Learning automata - A survey", *IEEE Trans. on SMC*, Vol.4, No.4, pp.323-334 (1974)
- (11) K.S. Narendera and M.A.L. Thathachar: *Learning Automata - An Introduction*, *Prentice-Hall International* (1989)
- (12) 銭 飛・平田廣則:「分散型強化学習システム: 学習オートマトンのチームモデル」, 電学論誌 C, **115**, 5, pp.787-793 (1998)
- (13) B. J. Oommen and J. K. Lanctôt: "Discretized Pursuit Learning Automata", *IEEE Trans. Syst. Man and Cybern.*, Vol.20, No.4, pp.931-938 (1990)
- (14) G. Karypis and V. Kumar: A Parallel Algorithm for multilevel graph partitioning and sparse matrix ordering, <http://www-users.cs.umn.edu/~karypis/>, *Journal of Parallel and Distributed Computing*, (1996)

池 坊 繁 屋 (正員) 1999 年広島国際学院大学 (旧広島電機大

学) 大学院工学研究科修士課程電子工学専攻修了。同年同大学大学院工学研究科博士課程計測制御専攻に入学, 現在に至る。学習オートマトンの基礎理論, 及びその応用の研究に興味を持つ。電子情報通信学会会員。

銭 飛 (正員) 1991 年千葉大学大学院自然科学研究科博士課程修了。同年広島国際学院大学 (旧広島電機大

学) 電子工学科講師, その後, 同情報工学科助教授を経て, 現在同教授。学習オートマトンの基礎理論, 強化学習システムの基礎理論、大規模システムのモデリング, 解析, 及びコンピュータネットワークなど並列分散システムの基礎的理論の研究に従事。Ph.D., IEEE, 情報処理学会, 計測自動制御学会, 電子情報通信学会会員。

平 田 廣 則 (正員) 1976 年東京工業大学大学院博士課程電気

工学専攻修了。同年千葉大学工学部助手, 1981 年同助教授, 1994 年同教授 1997 年千葉大学大学院自然科学研究科教授となり, 現在に至る。大規模システムのモデリング, 解析と設計, 特に生態システム, VLSI レイアウト, 並びに分散型システムの基礎的理論に興味を持つ。2001 年電気学会業績賞授賞。工学博士。IEEE, INNS, ISEM, 電

子情報通信学会, 計測自動制御学会, 情報処理学会, システム制御情報学会, 人工知能学会会員。