内部表現の冗長化によるボルツマンマシンの動作並列化

徐 丹[†] 熊沢逸夫[†]

ボルツマンマシンは，動作が並列に行われることができると一見思われているが，従来の一般的な枠組では，複数の素子の状態を同時に更新したときに，エネルギー最小状態が最大確率で発生することが理論的には保証されておらず，並列動作させた場合に，最適化問題を正しく解ける保証がない．しかしながら，実際に動作させてみると，数パーセントの素子を同時に状態更新してもうまくいく場合が多い．本論文では，勾配法の立場から，並列動作させた場合のボルツマンマシンの振舞いを解析し，原問題を変換した内部表現の冗長化により，並列動作を行っても確実にエネルギー最小状態の出現確率を最大とすることができることを示す．巡回セールスマン問題を例として，詳細なシミュレーションを行い，表現を冗長化した場合としない場合とのボルツマンマシンの振舞いに関する比較データを示す．この実験データから，内部表現の冗長化により，100% の素子を同時に状態更新してもエネルギー最小状態の出現確率を最大とすることができることが明らかになった．直観的にも，冗長表現の頑強性から，並列動作を行いやすくなることは予測されるが，本論文では，その直観的予測に対して，理論的，実験的裏付けを与えている．

Parallelization of Boltzmann Machine Using Redundant Internal Representation

DAN XU[†] and ITSUO KUMAZAWA[†]

If multiple units in the traditional Boltzmann Machine are updated simultaneously, the property that the minimum energy state will appear with the highest probability no longer holds. In this paper, a method using a redundant internal representation of the problem is shown for parallelizing the Boltzmann Machine. A simulation for the Traveling Salesman Problem using this method is shown. In our model, if multiple or even all units are updated simultaneously, the property that the minimum energy state will appear with the highest probability still holds, and the average calculating time for finding the minimum can be sharply reduced.

1. はじめに

ボルツマンマシンは一種のニューラルネットとして，一見並列化が容易であるかのように見え，それがボルツマンマシンの長所であるとも考えられているが，実際には，正しく並列動作させるために特別な工夫が必要であることが示されている[6]．

ボルツマンマシンは，熱平衡状態に達した後，エネルギー最小状態，すなわち，組合せ問題の最適解を表す状態を最も高い確率で発生する．この性質を利用して，問題の最適解を見つけ出すことができる．しかしながら，従来の方式では，素子の状態を逐次的に一つずつ更新していかなければ，この性質が保証されなかった．

ボルツマンマシンを使って組合せ最適化問題を解く場合，解くべき問題の評価関数がボルツマンマシンのエネルギー関数に一致するように，ボルツマンマシン中の各素子の閾値と結合強度を定める．この際に，問題をどのようにコーディングするか，すなわち，最適化問題中の各変数とボルツマンマシン中の素子の状態をどのように対応付けるかということには自由度がある．対応付けのルールは内部表現と呼ばれる．同じ問題に対して異なる内部表現を用いることによって，ネットワークの性質が大きく変わってくる[1),7)]．

ボルツマンマシンの問題には，本論文で扱う並列動作の問題のほかに，極小点の問題があることが良く知られている．すなわち，エネルギー最小状態を高頻度に発生させるために温度を低くしなければならないが，そのとき，ネットワークが極小点から抜け出すことが非常に難しくなるという問題である．文献 8) では，EBM ('Ensemble' Boltzmann Machine) と呼ばれるボルツマンマシンと Hopfield ネットワークの間に介在するネットワークが提案されている．EBM で

† 東京工業大学工学部情報工学科
Department of Computer Science, Faculty of Engineering, Tokyo Institute of Technology

は，従来のボルツマンマシン中の一つの素子に対して，多数の素子を用意し，それらの素子の平均値を用いて従来の一つの素子の状態を表す．それによって，極小点問題を改善できることが示されている．また，文献 9)，10) では，文献 8) と異なった視点から，解くべき問題の内部表現を通常より多い素子を用いて冗長化することにより，極小点問題を改善できる手法が提案されている．文献 8) と文献 9)，10) の違いは，前者では冗長化が素子別に行われているが，後者では複数の素子の状態をまとめて冗長化しているところにある．

本論文では，問題の内部表現の冗長化がボルツマンマシンの動作の並列化にも有効であることを示す．まず，前半では，従来のボルツマンマシンでは，並列動作がなぜできないかということを，勾配法の立場から分析する．

後半では，前半での分析の結果に基づき，問題の内部表現を冗長化することにより，同時に状態更新する素子の割合，つまり並列化の度合を高めても，ボルツマンマシンが近似的に正しく振舞い，正しい解を算出するようになることを示す．特に，代表的な組合せ最適化問題の一つである巡回セールスマン問題を例として，詳細なシミュレーションを行い，内部表現冗長化の並列動作における効果を検証する．このシミュレーションでは，実際的な問題において，並列度を高める上で有効となる具体的な工夫をいくつか提案する．また問題の内部表現を冗長化した場合と冗長化しない場合のボルツマンマシンの振舞いに関する詳細な実験データを示す．

シミュレーションの結果から次のことが明らかになった．(i)内部表現を冗長化すれば，全素子を同時に更新してもエネルギー最小状態の出現確率が最大となる．すなわち，他の局所最適解に落ち着く確率よりも高くなる．一方，従来方式のまま同時に複数の素子の状態を更新すると，並列度を高めるにつれ，エネルギー最小状態を得ることが困難になる．(ii)並列ハードウェアを用いればもちろんであるが，逐次型計算機上でのシミュレーションでも，内部表現の冗長化により，解を得るまでの計算時間を短縮できる．具体的には，本論文中の実験条件下で，$n=10$ 都市の巡回セールスマン問題に対して，

- 並列ハードウェアを用いた場合，エネルギー最小状態を見つけるまでに要する平均時間は，従来方式の約 1/50 にまで短縮される．ただし，表現を冗長化する前に，必要な素子数は $N=n\times n=100$ であったのに対して，表現を冗長化した後，必要な素子数は $M=12.8\times N=1280$ となる．
- 逐次方式のハードウェア上で計算する場合，本論文の方式では，冗長化によって素子数が増えた分，あるいは非線形関数の導入で計算が複雑となった分，計算量が増える．しかしながら，それ以上にエネルギー最小状態を見つけるまでの状態更新回数が大きく減るため，平均計算時間は，冗長化しない場合に比べて約2分の1に減る．

冗長化して素子数が増えているのにも関わらず，また逐次計算機上で計算しているのにも関わらず，エネルギー最小状態を見つけるのに要する平均計算時間が減る理由については本文中で考察する．

2. ボルツマンマシンとその並列動作

2.1 ボルツマンマシン

Hintonらにより提案されたボルツマンマシン(Boltzmann Machine)[4] は Hopfield の相互結合ニューラルネット[2),3)]から発展してきたものである．Hopfield ネットワークが単純な勾配法に基づいて動作するのに対して，ボルツマンマシンの動作には統計力学のアイデアが取り入れられ，素子の状態更新規則が非決定的になっている．充分状態遷移を繰り返した後，ネットワークは初期状態を忘却し，熱平衡状態に到達する．そこでエネルギー関数を最小にする状態の出現確率が最大になるためには，最も緩い条件として，遷移が許されている任意の2状態間でエネルギーを小さくする方向への遷移が，その逆方向の遷移に比べ高い確率で起こるようになっていることが不可欠である．特に状態 $\boldsymbol{x}_a$ から状態 $\boldsymbol{x}_b$ への遷移確率が次式

$$p(\boldsymbol{x}_a\rightarrow\boldsymbol{x}_b)=\frac{1}{1+\exp\dfrac{E(\boldsymbol{x}_b)-E(\boldsymbol{x}_a)}{T}} \qquad (1)$$

に従うならば，この条件は満足され，更に熱平衡における分布は，ボルツマン分布

$$P(\boldsymbol{x})=\frac{\exp\dfrac{-E(\boldsymbol{x})}{T}}{\sum\limits_{\boldsymbol{y}\in\Omega}\exp\dfrac{-E(\boldsymbol{y})}{T}} \qquad (2)$$

となる．ここで，$E(\boldsymbol{x})$ はエネルギー関数であり，T は正の定数で，温度と呼ばれる．また Ω はネットワークが取り得るすべての状態からなる集合である．Hinton らの提案したボルツマンマシンでは，一度に一つの素子しか状態を変化させることができないか

ら，ある状態から遷移可能な状態は元の状態からハミング距離が1離れたものに限られているが，より一般的には，式(1)で状態遷移を行うようにすれば，一度にハミング距離が離れた状態に遷移する場合にも熱平衡での分布は式(2)となることが分かっている☆.

ボルツマンマシンでは，一般には，各素子が0と1の2値をとり，エネルギー関数が次のような二次形式をとる.

$$E(\boldsymbol{x})=-\frac{1}{2}\sum_{i}\sum_{j\neq i}w_{ij}x_ix_j-\sum_{i}\theta_ix_i. \qquad (3)$$

ここで，x_i で素子 i の状態を，θ_i で素子 i の閾値を，w_{ij} $(w_{ij}=w_{ji})$ で素子 i と素子 j の間の結合強度を表す．このとき，式(1)によれば，ネットワークの状態更新は次のような規則によって行われる.

(i) N 個の素子の中の一つを等確率で選び，それを素子 i とする.

(ii) 他の素子の状態は一定としたまま素子 i の状態 x_i を0としたときと1としたときのエネルギー差を次式で計算する.

$$\begin{aligned}\Delta E_i&=E(\boldsymbol{x})|_{x_i=0}-E(\boldsymbol{x})|_{x_i=1}\\&=\sum_{j\neq i}w_{ij}x_j+\theta_i. \qquad (4)\end{aligned}$$

(iii) 次の確率で素子 i の状態を1とする.

$$p(x_i=1)=\frac{1}{1+\exp\dfrac{-\Delta E_i}{T}}. \qquad (5)$$

2.2 並列動作の問題

ボルツマンマシンでは，多数の素子が相互結合されており，ハードウェア的には通信や状態更新を並列に行う能力があるが，従来のボルツマンマシンの理論では，素子の状態更新は，一つずつ逐次的に行わなければならず，並列に状態更新することができない．従来の方式で，無理に，複数の素子を同時に状態更新すると，エネルギーを小さくする方向へ遷移する確率が，その逆の大きくする方向に遷移する確率より高くなることが保証されなくなり，エネルギー最小点の出現確率が最大になる性質は失われてしまう[6].

この辺の事情を少し詳しく見てみよう.

ネットワークが状態を更新するとき，エネルギー関数の三次以上の偏微分が全部0となることを注意し，テーラの定理を用いると，エネルギー変化は次式で表される.

$$\begin{aligned}\Delta E&=\Big(\sum_{i=1}^{N}\Delta x_i\frac{\partial}{\partial x_i}\Big)E(\boldsymbol{x})+\frac{1}{2}\Big(\sum_{i=1}^{N}\Delta x_i\frac{\partial}{\partial x_i}\Big)^2E(\boldsymbol{x})\\&=\sum_{i=1}^{N}\Delta x_i\Big(-\sum_{j\neq i}w_{ij}x_j-\theta_i\Big)\\&\quad-\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}w_{ij}\Delta x_i\Delta x_j. \qquad (6)\end{aligned}$$

ここで，Δx_i は素子 i の状態が更新した後と更新する前の状態の差である．一時点で，状態を更新する素子が一つしかない場合，それを素子 i とすると，エネルギー変化は

$$\Delta E=\Delta x_i(-\sum_{j\neq i}w_{ij}x_j-\theta_i) \qquad (7)$$

となる．これを式(1)に代入すると，式(4)(5)のような逐次方式の場合の状態更新規則が得られる．すなわち，この場合には，結線で結ばれた素子の状態だけを見て状態を更新できる．それに対して，複数の素子が同時に状態を更新する場合には，エネルギー最小点の出現確率を最大にするために，式(1)により状態遷移を決めなければならないが，このときは，$\Delta E=E(\boldsymbol{x}_b)-E(\boldsymbol{x}_a)$ を式(6)で計算しなければならず，これはもはや結線で結ばれた素子の状態のみを使って局所的に計算するというニューラルネットの枠組から外れてしまう.

2.3 勾配法からみた従来のボルツマンマシンの並列動作

式(6)を考える．状態が変わっていない場合，$\Delta x_i=0$ となる．状態が変わった場合，$\Delta x_i=\pm1$ となる．素子 i の更新した後の状態を $\bar{x}_i$ とすると，式(6)の第1項中の各成分は，

$$\begin{aligned}\Delta x_i\frac{\partial E}{\partial x_i}&=\Delta x_i(-\sum_{j\neq i}w_{ij}x_j-\theta_i)\\&=(\bar{x}_i-x_i)(-\sum_{j\neq i}w_{ij}x_j-\theta_i)\\&=E(\boldsymbol{x})|_{\bar{x}_i}-E(\boldsymbol{x})|_{x_i} \qquad (8)\end{aligned}$$

となる.

式(8)の中の $E(\boldsymbol{x})|_{\bar{x}_i}$ と $E(\boldsymbol{x})|_{x_i}$ はそれぞれ他の素子の状態を一定としたまま，x_i の状態を更新した後と更新する前のエネルギー値である．したがって，$\Delta x_i\dfrac{\partial E}{\partial x_i}$ は，他の素子の状態を一定としたまま，素子 i の状態のみを更新することによるエネルギー変化を表す．複数の素子が同時に式(5)によって状態を更新する場合，各素子はそれぞれ0と1のうち，$\Delta x_i\dfrac{\partial E}{\partial x_i}$ をより小さくする方へ，より高い確率で更新される．したがって，K 個の素子が同時に状態を更新する場合，ネットワークは，2^K 個の異なった状態へ遷移す

☆ 例えば，文献1) の p. 30 の証明はエネルギー関数の具体的形状や一度に状態更新する素子数に依存しない一般的なものになっている.

ることが可能であるが，その中で $\sum_{i=1}^{N} \Delta x_i \frac{\partial E}{\partial x_i}$ を小さくする状態にほど，高い確率で遷移する．つまり第1項に関しては，エネルギー値を小さくする方向へ変わるが，このように状態が変わるとき，$\Delta x_i = \pm 1$ となり，x_i の変化量が大きくなるため，式(6)の中の第2項が無視できない．したがって，全体としてみたときに，式(6)の ΔE を小さくする状態（すなわち，より小さいエネルギーを持つ状態）へ，より高い確率で遷移することが保証できなくなる．つまり，熱平衡状態でエネルギー最小点の出現確率が最大になる性質は保証されなくなってしまう．

実際に，すべての素子の状態が同時に状態を更新する完全並列方式 (Unlimited Parallelism) という特別な場合について，文献 6) で，その平衡分布が求められ，熱平衡状態でエネルギー最小状態の出現確率が一番大きくなる性質が失われることが証明されている．

文献6) では，その他に，制限付き並列方式(Limited Parallelism) という特別な場合が考察されている．制限付き並列方式では，「各部分集合中の素子が互いに結合を持たない」という条件を満たすように，全素子の集合をいくつかの部分集合に分ける．このようにすると，ある部分集合中の一つの素子の状態更新は同じ部分集合中の他の素子の状態更新に影響を与えないから，部分集合ごとにその中のすべての素子を同時に状態更新しても，式(6)の第2項は0となる．したがって，ネットワークは常にエネルギーのより小さい状態へより高い確率で遷移することになる．明らかに，逐次方式のボルツマンマシンは，この部分集合の要素数が1である特殊な場合となっている．

一般の問題で，素子の間に密度の高い結合がある場合，このような部分集合への分割は難しい．特に完全グラフ的に結合がある場合，部分集合への分割は全く行うことができない．しかし，もしも結線トポロジーが特殊な形状をしており，このような部分集合に分割可能であれば，上に述べたような部分的な並列化が可能であり，並列化してもエネルギー最小状態の出現確率が一番大きくなる性質が保証される．

3. ボルツマンマシンの動作並列化における内部表現冗長化の効果

この章では，文献 9)，10) で提案された内部表現の冗長化手法を用いることによって，**2.** で述べた，ボルツマンマシンの動作並列化に当たって生じる問題点を克服し，並列動作を行ってもエネルギー最小点の出現確率を最大とすることができることを直観的に示す．

3.1 冗長表現の定義

ネットワーク上で情報を表現する方法を大きく分けると，局所的表現と分散的表現の二種類がある[1]．局所的表現では，一つの表現したい項目を一つの素子の状態で表す．それに対して，分散的表現では，一つの項目はいくつかの素子の状態の組合せとして表される．逆に言えば，分散表現では一つの素子の状態がいくつかの項目の表現に関わっている．局所的表現と比べて，分散的表現の方が，正しい解が得られる機会が増えることと，誤った計算に対する頑健性があることなど，いろいろな利点があると指摘されている[1),7)]．

冗長表現は，分散的表現の一種である．問題の内部表現に適当な冗長性を与えることにより，ボルツマンマシンの極小点問題を解決できる可能性があることが指摘されている[8)~10)]．ここでは，冗長表現が並列動作にも適した表現になっていることを示す．

文献10) で使われた内部表現の冗長化方法を基にして，次のような一般的な冗長化方法を考える．

N 個の素子 $\boldsymbol{x}=(x_1, x_2, \cdots, x_N)^T$ のネットワークに対して，それを冗長化した $M(M>N)$ 個の素子 $\boldsymbol{y}=(y_1, y_2, \cdots, y_M)^T (y_i \in \{1, -1\},\ i=1, 2, \cdots, M)$ (以前の説明では素子の状態を 0, 1 としたが，以下便宜上 -1, 1 とする．こうしてもボルツマンマシンの本質に変化はない) から成るネットワークを考える．$\boldsymbol{y}$ と $\boldsymbol{x}$ を次のように対応付ける．

$$\boldsymbol{z}=\frac{1}{K}Q\boldsymbol{y}. \tag{9}$$

$$\boldsymbol{x}=F(\boldsymbol{z})=(f(z_1), f(z_2), \cdots, f(z_N))^T. \tag{10}$$

ただし，

K：定数であり，冗長度を表す．

Q：$N \times M$ の行列である．$q_{ij} \in \{1, 0, -1\}$，$i=1, 2, \cdots, N$；$j=1, 2, \cdots, M$.
Q の各行の中に0でない成分が K 個ある．

$\boldsymbol{z}$：$\boldsymbol{z}=(z_1, z_2, \cdots, z_N)^T$，$-1 \le z_i \le 1$，$i=1, 2, \cdots N$.

$f(z)$：$[-1, 0]$ で単調減少で，$[0, 1]$ で単調増加の可微分な連続関数である．しかも，$f(0)=0$，$f(-1)=f(1)=1$ とする．

こうすれば，元のネットワークの一つの素子 x_i の状態が表す情報は冗長化したネットワークでは K 個の素子 $y_j (q_{ij} \neq 0)$ の状態を使って分散表現される．また一つの y_j の状態が複数の x_i の状態に関わる．この写像により $\boldsymbol{y}$ に対するエネルギー関数 $E(\boldsymbol{y})$ を $\boldsymbol{x}$ に対するエネルギー関数 $E(\boldsymbol{x})$ から得ることができ，$\boldsymbol{x}$

空間でエネルギー最小点を求める問題は，$\boldsymbol{y}$ 空間でエネルギー最小点を求める問題と等価になる．

3.2 冗長化表現を用いた場合の並列動作の可能性

ネットワーク $\boldsymbol{y}$ で複数の素子が同時に状態を更新したときの各 z_i の変化量は次式になる．

$$\Delta z_i = \sum_{j=1}^{M} \frac{1}{K} q_{ij} \Delta y_j. \tag{11}$$

冗長度 K が十分大きい場合，同時に状態を更新する素子の割合を小さくすることで，各 $|\Delta z_i|$ の値を十分小さくすることができる．また同時に多数の素子を状態更新したとしても，もし各素子の状態変化が協調して $|\Delta z_i|$ を大きくする方向に生じるのでなければ，やはり $|\Delta z_i|$ の値を十分小さくすることができる．このとき，次の近似式が成り立つ．

$$\begin{aligned}\Delta x_i &\approx \frac{\partial x_i}{\partial z_i}\Delta z_i \\ &= \frac{\partial x_i}{\partial z_i}\sum_{j=1}^{M}\frac{1}{K}q_{ij}\Delta y_j \\ &= \frac{\partial x_i}{\partial z_i}\sum_{j=1}^{M}\frac{\partial z_i}{\partial y_j}\Delta y_j \\ &= \sum_{j=1}^{M}\frac{\partial x_i}{\partial y_j}\Delta y_j. \end{aligned}\tag{12}$$

関数 f は連続関数であるから，このとき，各 $|\Delta x_i|$ の値も小さくなる．

2.3 で説明したように，ボルツマンマシンで並列動作を行う場合，エネルギー関数の各変数 x_i の変化量が大きいために，式(6)の二次微分項の影響が強く現れ，エネルギー関数が小さくなる方向へ状態遷移が起こる確率がその逆方向へ遷移する確率に比べて高くなることが保証できなかった．式(6)を考えると，この二次微分項を無視できるようにするためには，各 x_i の変化量 Δx_i の絶対値を小さくすることができればよいわけで，もしも上の前提（各素子の状態変化が協調して $|\Delta z_i|$ を大きくする方向には起こらない）が成り立つならば，内部表現を冗長化することにより，同時に複数の y_j を状態更新しても各 x_i の変化量の絶対値を小さくすることができ，エネルギー最小状態の出現確率を最大にできることが分かる．

同時に状態を更新する素子の割合が大きくない場合，あるいは，各素子の状態変化が協調して $|\Delta z_i|$ を大きくする方向には起こらない場合，Δx_i は微小量であるから，式(6)の二次微分項を無視できるようになり，エネルギー関数の変化量を次式で近似できる．

$$\begin{aligned}\Delta E &\approx \sum_{i=1}^{N}\frac{\partial E}{\partial x_i}\Delta x_i \\ &\approx \sum_{i=1}^{N}\frac{\partial E}{\partial x_i}\sum_{j=1}^{M}\frac{\partial x_i}{\partial y_j}\Delta y_j \\ &= \sum_{j=1}^{M}\left(\sum_{i=1}^{N}\frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial y_j}\right)\Delta y_j \\ &= \sum_{j=1}^{M}\frac{\partial E}{\partial y_j}\Delta y_j. \end{aligned}\tag{13}$$

ΔE が負の値をとる確率を高くするためには，同時に状態を更新する各素子が ± 1 のうち，$\frac{\partial E}{\partial y_j}\Delta y_j$ の値をより小さくする方に状態を更新する確率を高くすればよい．すなわち，

- $\frac{\partial E}{\partial y_j} > 0$ のときには，y_j が -1 になる確率が高い，
- $\frac{\partial E}{\partial y_j} < 0$ のときには，y_j が 1 になる確率が高い，

となるように，y_j を時間変化させれば良い．そのために，次のような状態更新規則を用いれば良い．

$$P(y_j=1) = \frac{1}{1+\exp\dfrac{\partial E/\partial y_j}{T}}. \tag{14}$$

以上より，内部表現 $\boldsymbol{x}$ を直接並列に状態更新することはできないが，それを適当な写像で変換して冗長化した内部表現 $\boldsymbol{y}$ は，式(14)で並列に状態更新できることが分かる．

よく知られているように，素子の状態が $[0,1]$ 上の連続値を取る Hopfield のネットワークでは，一度の状態変化が微小であるならば並列に状態更新することができる[2]．この事実から，上記の結果をわかりやすく説明するならば，素子の状態が 0 と 1 の二値を取るボルツマンマシンでも，連続的に（微小量で）変化し得る媒介的な変数を介在してエネルギー関数が表現されていれば並列動作ができるということになる．

以上の結果をまとめると，次の手順でボルツマンマシンの動作を並列化できる．

（ⅰ） 問題の表現方法を **3.1** で述べたように冗長化する．

（ⅱ） 同時に状態を更新する各素子の状態変化が協調して $|\Delta z_i|$ を大きくする方向に起きることのないように工夫をする．

（ⅲ） 各時点で複数の素子を選び，式(14)に従って同時に状態を更新する．

以下の章で，巡回セールスマン問題を例として，（ⅱ）の条件が成り立つようにするための具体的な工夫と（ⅲ）の複数の素子の選び方について論ずる．

4. 巡回セールスマン問題に対する内部表現の冗長化方法

n 都市の巡回セールスマン問題をニューラルネットで解く場合，次のような内部表現がよく使われる．

n 行 n 列の行列状に並んでいる n^2 個の素子を考える．各列はそれぞれ各都市に対応し，各行は各都市をまわる順番に対応する．こうすると，例えば，$A_2 \rightarrow A_1 \rightarrow \cdots \rightarrow A_n \rightarrow A_2$ の経路を**図1**に示す行列で表現できる．意味のある順路を表現する行列には，必ず，各行列に1が一つだけあるということに注意する必要がある．この n 行 n 列の行列を $\boldsymbol{x}$ で表す．その列ベクトルをそれぞれ $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n$ で表す．

このような表現を用いると，巡回セールスマン問題を次式の目的関数の最小化として定式化することができる．

$$E_H(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\neq i}^{n}\sum_{k=1}^{n} d_{ij}x_{ki}(x_{k-1,j}+x_{k+1,j}) + A\left(\sum_{i=1}^{n}\left(\sum_{k=1}^{n} x_{ik}-1\right)^2 + \sum_{k=1}^{n}\left(\sum_{i=1}^{n} x_{ik}-1\right)^2\right). \quad (15)$$

ただし，d_{ij} は都市 A_i と A_j の間の距離で，x_{ik} は i 行目 k 列目の素子の状態である．正の定数 A を十分大きくすると，関数 $E_H(\boldsymbol{x})$ の最小点は最短経路に対応するようになる．

ここで，文献10)で提案された次のような冗長化方法を使う（紙面の都合上，この方法の概略だけを与えておく．詳しくは文献10)を参照してほしい）．

$$\boldsymbol{z} = \frac{1}{m}Q\boldsymbol{y}. \quad (16)$$

$$\boldsymbol{x} = F(\boldsymbol{z}) = (f(z_{ij})). \quad (17)$$

ただし，

1. $\boldsymbol{y}$ は m 行 n 列の行列で，その各成分が ± 1 の二値をとる．その各列をそれぞれ $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n$ で表す．
2. Q は n 行 m 列の行列で，その各成分も ± 1 の二値をとる．Q の各行ベクトルは互いに直交する．その行ベクトルを $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_n$ で表す．
3. $\boldsymbol{z}$ は n 行 n 列の行列で，その各成分は $[-1, 1]$ 上の値をとる．
4. f を次式に示す関数とする（**図2**）．

$$f(z) = 1-(1-z^2)^2. \quad (18)$$

すなわち，$\boldsymbol{x}$ を列ベクトルごとに独立に冗長化している．その第 j 列 $\boldsymbol{x}_j$ を冗長化したものが $\boldsymbol{y}_j$ になっている．したがって，$\boldsymbol{x}$ の n 個の素子が表す情報を $\boldsymbol{y}$ の m 個の素子の状態で表すことになる．

以上に述べた $\boldsymbol{y}$ から $\boldsymbol{x}$ への写像を，まとめて $\boldsymbol{x} = F\left(\frac{1}{m}Q\boldsymbol{y}\right)$ と表すことにする．明らかにこの冗長化の写像は，**3.** で述べた一般的写像の特殊な場合になっている．

このように冗長表現 $\boldsymbol{y}$ を得るとそれは次のような性質を持つ[10]．

（i） $\boldsymbol{y}_j$ が $\boldsymbol{q}_i$ あるいは $-\boldsymbol{q}_i$ と一致するとき，また，そのときに限り，$\boldsymbol{x}_j$ は $\boldsymbol{e}_i$ となる．ここで，$\boldsymbol{e}_i$ は i 番目の成分だけが1で，ほかの成分が0となるような縦ベクトルである．したがって，表現 $\boldsymbol{y}$ では，n 個の列ベクトル $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n$ がそれぞれ $\pm\boldsymbol{q}_i$ ($i=1, 2, \cdots, n$) の中の異なるベクトルをとるとき，表現 $\boldsymbol{x}$ では，各行各列に1が一つだけ生じるようになる．このときに $\boldsymbol{y}$ は一つの意味を持った経路を表す．

（ii） 各 $\boldsymbol{y}_j$ が $\boldsymbol{q}_i$ に完全に一致しなくても，その近くにあれば，$\boldsymbol{q}_i$ に相当する結果が表されていると考える．こうすることで，多数の状態が一つの経路を表すことになる．

（iii） 図2に示される形状の関数を f として用いるのは次のような理由による．上述したように，各 $\boldsymbol{y}_j$ がそれぞれ異なる $\pm\boldsymbol{q}_i$ の近くにあるとき，$\boldsymbol{y}$ は一つの経路を表すことになる．このとき，式(16)により求まる z_{ij} は0あるいは ± 1 に近い値を持つが，それを図2に示される関数 f によって変換すると，各 x_{ij} は

	A_1	A_2	$\cdots$	A_N
1	0	1	$\cdots$	0
2	1	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
N	0	0	$\cdots$	1

図1 Hopfield の巡回セールスマン問題の内部表現
Fig. 1 The internal representation of the TSP used by Hopfield.

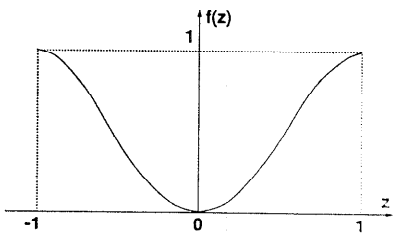


図2 関数 $f(z)$ の形状
Fig. 2 The function $f(z)$.

更に0あるいは1に近くなる．これにより，何らかの経路を表す状態（意味のある状態）のエネルギー関数値が小さくなり，その出現確率は大きくなる．

以上の方法で得られた冗長表現 y に対するエネルギー関数は式(15)に示すエネルギー関数 $E_H(\boldsymbol{x})$ に $\boldsymbol{x}=F\left(\frac{1}{m}Q\boldsymbol{y}\right)$ を代入して得られる $E_H\left(F\left(\frac{1}{m}Q\boldsymbol{y}\right)\right)$ となる．

5. 高度な並列性を得るための工夫

並列度（同時に状態を更新する素子の割合）を向上するために，上述の内部表現を使うほかに，いくつかの工夫を導入する．それは，同時に状態更新する素子集団の決め方，エネルギー関数の構成，および温度値の選び方に関する工夫である．以下でこの三つの工夫を見て行くことにする．

5.1 温度の決め方

状態遷移に影響を及ぼすパラメータに温度がある．温度 T は素子の状態更新の不確定性を決めるパラメータで，その値を大きくすると，素子の動作はよりランダムになり，各素子の状態が協調して $|\Delta z_i|$ を大きくする方向に変化することは起こりにくくなると考えられる．つまり，式(11)によると，冗長度 K が大きい場合，Δy_i のランダム性が高いと Δz_i の値は0に近くなり，並列化しやすくなる．このように，並列化のためには温度が高い方が良い．このことは温度を低くすること，つまり，焼きなまし[5]を行うことができなくなることを意味しているが，文献9）に示されているように，内部表現を冗長化した場合，温度が高くとも，エネルギー最小点の出現確率を高くすることができ，焼きなましを行う必要がなくなるから，このことによって不利になることはない．実際，後半のシミュレーションでは，温度は比較的高く設定し，一定としているが高い確率で解を得ることに成功している．

5.2 同時に状態更新する素子集団の決め方

全素子の中からある割合で選ばれた素子を同時に状態更新する場合，すべての $|\Delta z_{ij}|$ をできるだけ小さくするためには，同時に状態を更新する素子が一つの特定の z_{ij} に関係ある素子に集中しないようにすると良い．このために，各時点で，全素子の中から状態更新を行う素子集団 U_s を次の手続きで決める．毎時点で，各素子独立にそれが U_s に含まれる確率を g $(0<g\leq 1)$ とする．こうすると，毎時点，全素子の内の約 $100g$ パーセントの素子が同時に状態を更新することになる．また，U_s は全くランダムに決めているから，ある z_{ij} の値に関係ある m 個の素子の中の約 gm 個の素子が同時に状態を更新する．このとき，統計的に考えると，各 z_{ij} の変化量の絶対値は，$2g$ 以下に抑えることができ，同時に状態更新する素子が特定の一つの z_{ij} に関係ある素子に集中することは非常に低い確率でしか起こらない．言い替えれば，エネルギー最小状態が最大頻度で出現することを保証するために，z_{ij} の最大変化量が δ に抑えられていなければならない場合には，$g=\delta/2$ とすることで，各 z_{ij} の許される変化量の下で，最大の並列度を得ることができる．

以上の素子選択のスケジューリングは，各時点で各素子ごとにランダムにそれを状態更新するかどうかを決めるだけなので極めて単純であり，ハードウェアの単純化のためにも都合良い．

5.3 エネルギー関数への $|\Delta z_{ij}|$ を小さくする拘束の導入

エネルギー関数の構成について検討する．次のような関数を考える．

$$E_A(\boldsymbol{y})=-\sum_{i=1}^{n}\sum_{j=1}^{n}z_{ij}^2. \tag{19}$$

先に示した E_H に上式に適当な重み B を付けた項を加え，実際に用いるエネルギー関数を次式とする．

$$E(\boldsymbol{y})=E_H\left(F\left(\frac{1}{M}Q\boldsymbol{y}\right)\right)+BE_A(\boldsymbol{y}). \tag{20}$$

B は正数とする．こうして得られたエネルギー関数の最小点は，依然として最短経路に対応する[10]．

第2項は次のような働きをする．

$E_A(\boldsymbol{y})$ の中で，行列 $\boldsymbol{z}$ の第 j 列の値に関係ある部分を抜き出すと次式になる．

$$\begin{aligned}E_j&=-\sum_{i=1}^{n}z_{ij}^2\\&=-\frac{1}{m^2}\sum_{i=1}^{n}\boldsymbol{q}_i\boldsymbol{y}_j\boldsymbol{q}_i\boldsymbol{y}_j\\&=-\frac{1}{m^2}\boldsymbol{y}_j^T\left(\sum_{i=1}^{n}\boldsymbol{q}_i^T\boldsymbol{q}_i\right)\boldsymbol{y}_j.\end{aligned} \tag{21}$$

これはちょうど連想記憶ネットワークのエネルギー関数と同じ形になっている[3]．連想記憶ネットワークでは，複数の素子を同時に状態更新する場合に次の性質が成り立つことが知られている．

（i） 現在の状態 $\boldsymbol{y}_j$ がある $\boldsymbol{q}_i$ の近くにあるとき，z_{ij} の値は1に近い．このとき，複数の素子を同時に状態更新すると各素子は強く $\boldsymbol{q}_i$ の近くに引かれる．したがって，z_{ij} の値は更に1に近づくが，もともと z_{ij} の値は1に近かったか

ら，z_{ij} の変化は小さい．

（ii） 現在の状態 $\boldsymbol{y}_j$ が複数の $\boldsymbol{q}_i$ から等距離にある場合，z_{ij} の値は0に近い．このとき複数の素子を同時に状態更新すると，どの $\boldsymbol{q}_i$ へも弱くしか引かれないため，それらはほとんど無秩序ランダムに次の状態を決める．したがって，z_{ij} の値は0に近いままであり，この場合にも z_{ij} の変化は小さい．

（iii） 一般に，確率的状態更新規則を用いる連想記憶ネットワークでは，z_{ij} の値が0に近いときは，各素子に強い作用が働かず，振舞いはランダム無秩序となる．z_{ij} の値が1に近づくにつれてアトラクタへの引き込み作用が強く働き，動作が秩序だってくる．したがって，z_{ij} の値に関わる複数の素子を同時に状態更新しても，z_{ij} の値が一度に0から1へ，あるいはその逆へ大きく変化することは起り得ない．

以上の性質を持った第2項をエネルギー関数に加えることで，各 z_{ij} $(i=1,2,\cdots,n)$ の変化量が小さく保たれることがわかる．B の値が大きければ，この効果は顕著に現れる．一般には，並列度が大きくなるにつれて，B を適当に大きくする必要がある．

6. シミュレーション

10都市の巡回セールスマン問題を例として実験を行い，本論文で提案した並列化手法と従来手法の比較を行う．

6.1 実験方法

10都市の巡回セールスマン問題に対して，各 $\boldsymbol{y}_j$ の長さ m を128とし，全部で1280個の素子を用いた．行列 Q の構成法は文献10）と同じである．

同時に状態を更新する素子の選択は**5.2**に述べた方法に従って行う．

計算量を減らすために，式(14)によってではなく，次のように y_{kj} の状態を更新する．

- $-\dfrac{\partial E}{\partial y_{kj}}+\lambda\varepsilon\geq 0$ のとき y_{kj} を1にする．
- $-\dfrac{\partial E}{\partial y_{kj}}+\lambda\varepsilon<0$ のとき y_{kj} を -1 にする．

ここで，ε は期待値が0，標準偏差が1である正規乱数である．この方法で得られる確率分布は，式(14)によるものとほぼ同じになることが知られている[11]．λ を大きくすればするほど不確定性が大きくなる．式(14)の代わりにこの方法で確率的な動作を実現する場合，指数計算が必要でなくなり，シミュレーションの速度を上げることができる．

また，$\boldsymbol{y}_j$ とある $\boldsymbol{q}_i$ とのハミング距離が，$0.3m$ より小さくなり，他の $\boldsymbol{q}_k$ $(k\neq i)$ とのハミング距離が $0.3m$ 以上になったとき，$\boldsymbol{y}_j$ は $\boldsymbol{q}_i$ に一致したと考える．

冗長化する前のボルツマンマシンについての比較実験においても，上述と同じ理由で，確率的な動作は式(5)の代わりに次のように実現する．

- $\Delta E_i+\lambda\varepsilon\geq 0$ のとき y_{kj} を1にする．
- $\Delta E_i+\lambda\varepsilon<0$ のとき y_{kj} を -1 にする．

ここで，ΔE_i は式(4)で定義されたものである．

6.2 実験結果と考察

二つの都市配置を対象として実験を行った．それぞれの座標値を**表1**に示している．これらの都市配置に対する最短経路は**図3**に示されている．

まず，表現を冗長化せず，表現 x に対して，従来の状態更新規則を逐次的に適用した場合と，無理やり並列に適用した場合の動作を調べてみた．以後，前者を逐次的更新，後者を並列更新と呼ぶ．式(15)のエネルギー関数を用い，A と λ の最適の値は試行錯誤によって見いだした．具体的には，それぞれ4.2と2.5にした．同時に状態更新する素子の選び方（スケジューリング）として，前章に説明した方法を用いた．すなわち，各時刻に各素子が状態更新を行う確率を g とし，同時に約 $100g$ パーセントの素子が状態更新を行うようにする．配置Iについて，逐次的更新（表中では $g=0$ と記す）および $g=0.1, 0.2, \cdots, 0.9, 1.0$ の各場合の並列更新において，システムをそれぞれ1000000回動作させて，最小点の出現状況を調べてみた．ここで，逐次更新では1個の素子を状態更新したときそれを1回と数え，並列更新では一度に選ばれた約 $100g$ 個の素子の状態更新をしたときそれを1回として数えている．つまり仮想的な並列ハードウェア上のクロッ

表1 各配置の都市の座標値
Table 1 The coordinates of the cities of every placement.

都市番号	配置I	配置II	配置III
1	(2.0,1.5)	(0.0,0.0)	(1.0,0.0)
2	(8.0,3.0)	(2.0,0.0)	(3.0,0.0)
3	(6.5,2.0)	(4.0,0.0)	(4.0,0.0)
4	(7.0,5.0)	(0.0,2.0)	(0.0,1.5)
5	(1.5,2.0)	(1.3,2.0)	(1.5,2.0)
6	(2.5,7.5)	(2.6,2.0)	(2.5,1.5)
7	(4.0,4.5)	(4.0,2.0)	(4.0,1.5)
8	(9.0,6.0)	(0.0,4.0)	(1.3,3.0)
9	(5.0,9.0)	(2.0,4.0)	(3.0,3.0)
10	(6.0,2.2)	(4.0,4.0)	(6.0,2.2)

クタイムを基準として計っている．以後，このように数える回数の単位をサイクルと呼ぶことにする．結果は**表2**に示されている．表中の回数の総和は 1000000 とならないが，経路として意味を持たない状態が現れた場合を除外しているためこのようになっている．この結果をみると，g が非常に小さい場合，並列更新は逐次更新に似た状態の分布をとる．g の値が大きくなるにつれて，並列更新の動作は次第に逐次更新と異なったものになる．$g=0.5$ では，エネルギー最小状態の出現確率が最大となる性質が失われている．また $g=0.8$ 以上では，もはや意味のある経路に対応する状態は観測されない．

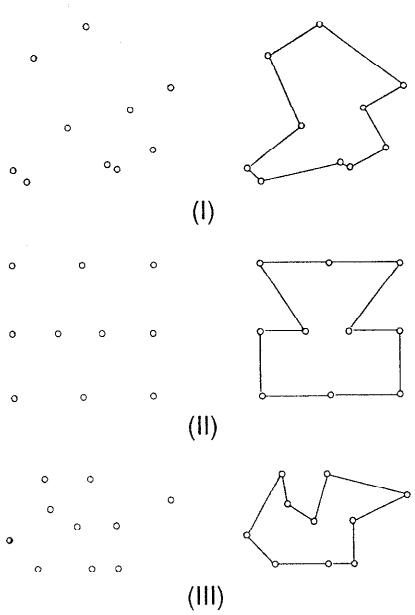


図 3　都市の配置とその最短経路．左側は都市の相対位置で，右側は最短経路である．
Fig. 3　The relative placements of the cities (left) and the correspondent shortest paths (right).

表 2　従来方式において，各並列度でネットワークを 1000000 サイクル動作させた時の各経路に対応する状態の出現回数
Table 2　The numbers that the states corresponding to paths appeared when the traditional model was performed 1000000 cycles using different g(pracement I).

経路の長さ	26.38	26.78	26.90	他の経路
g=0	885	566	442	15872
g=0.1	600	524	374	14429
g=0.2	267	245	337	10733
g=0.3	168	138	136	6365
g=0.4	111	44	41	2811
g=0.5	14	47	15	872
g=0.6	5	0	4	176
g=0.7	2	0	0	30
g=0.8	0	0	0	0
g=0.9	0	0	0	0
g=1.0	0	0	0	0

更に，g の各値について，システムを異なる初期状態から，それぞれ 50 回動作させて，最初に最小状態を見つけるまでのサイクル数を調べてみた．結果は**表3**に示されている．$g=0$ は逐次的方式の場合を示す．Max と Min はそれぞれ最小点を一番遅く見つけた場合と一番早く見つけた場合のサイクル数である．Mean は最小点を見つけるまでの平均サイクル数であり，50回の試行の平均をとったものである．

次に，前章で説明した冗長化方式を用いて並列更新を行う場合について調べてみた．まず，配置Iについて，並列度を $g=0.01, 0.02, 0.03, 0.04, 0.05, 0.10, 0.15, \cdots, 0.95, 1.00$ の各値にし，エネルギー関数の各部分の重み A, B と温度に相当する λ の値を変えながら実験した．それぞれ 50 の異なる初期状態から出発して何サイクル目にエネルギー最小状態を見出すかということを調べたものが**表4**である．

先に述べたように，複数の素子を同時に状態を更新するときに最小点が最も高頻度に出現するためには，$|\Delta z_{ij}|$ の値を小さくすることが必要である．この条件を充足するために，表4では，g の値を大きくするにつれて，B と λ の値も大きくしている．

表5には，$g=0.05, 0.25, 0.50, 0.75, 1.00$ の五つの場合にそれぞれネットワークを並列に 50000 サイクル動作させたときの各経路に対応する状態の出現回数を示す．明らかに，より短い経路に対応する状態の出現回数がより多くなっている．エルゴード性を考えると，同様の実験結果が初期状態を変えた場合にも得られるはずである．表2に比べて，はるかに高頻度にエ

表 3　従来方式において，各並列度で最短経路を見いだすまでの最大 (Max)，最小 (Min)，平均 (Mean) サイクル数
Table 3　The maximum, minimum, and mean numbers of cycle that the shortest path was found firstly when the traditional model was performed using different g(placement I).

g	Max	Min	Mean
0	1135022	12530	266022.3
0.1	131143	977	25305.8
0.2	87526	39	24799.6
0.3	152111	82	33011.5
0.4	268362	3006	53992.4
0.5	1406764	1270	207500.3

表 4 冗長化したネットワークで，配置Ⅰに対して，各並列度で最短経路を見いだすまでの最大 (Max)，最小 (Min)，平均 (Mean) サイクル数
Table 4 The maximum, minimum, and mean numbers of cycle that the shortest path was found firstly when the model with a redundant representation was performed using different g(placement Ⅰ).

A	B	λ	g	Max	Min	Mean
5.0	3.0	0.040	0	474779	11484	45239.5
5.0	3.1	0.042	0.01	11564	607	2680.2
5.0	3.2	0.044	0.02	10773	504	1685.7
5.0	3.3	0.046	0.03	3900	121	960.2
5.0	3.4	0.048	0.04	2763	115	903.5
5.0	3.5	0.050	0.05	2186	75	741.7
5.0	4.0	0.060	0.10	1968	137	486.2
5.0	4.5	0.070	0.15	1324	84	498.4
5.0	7.25	0.090	0.20	2360	116	771.2
5.0	10.0	0.110	0.25	2068	63	627.7
5.0	10.4	0.114	0.30	2315	59	547.9
5.0	10.8	0.118	0.35	1629	112	563.5
5.0	11.2	0.122	0.40	2109	88	624.4
5.0	11.6	0.126	0.45	2423	53	513.2
5.0	12.0	0.130	0.50	2271	33	560.7
5.0	12.4	0.134	0.55	2044	42	578.7
5.0	12.8	0.138	0.60	2647	15	587.4
5.0	13.2	0.142	0.65	2202	56	493.3
5.0	13.6	0.146	0.70	1867	65	488.3
5.0	14.0	0.150	0.75	2979	68	545.2
5.0	14.4	0.156	0.80	1859	38	497.4
5.0	14.8	0.162	0.85	2658	21	535.4
5.0	15.2	0.168	0.90	1630	34	492.4
5.0	15.6	0.174	0.95	1675	114	658.5
5.0	16.0	0.180	1.00	3692	76	921.7

ネルギー最小状態が得られていることが分かる．

エネルギー最小点を見つけ出す速度を見てみると，従来方式で無理に並列に状態更新した場合，表3によると最も早い場合でも最小点を見いだすのに平均24799.6 サイクルを必要としている．一方，本論文で提案した並列化方式では，最も早い場合わずか平均486.2 サイクルで最小点を見つけ出すことに成功している．つまり並列ハードウェアを使った場合，最小点を発見するまでの時間は約 1/50 に短縮される．これは，表現冗長化のために 10 倍の素子を使っていることを差し引いても余りある高速化である．これより，逐次方式の計算機上でシミュレーションした場合にも，計算時間が短縮されることが期待できる．実際に比較してみると，必要な平均計算時間は，従来方式で，$g=0.1$ の場合に一番速く，199.7 秒であるのに対して，本論文で提案した並列化方式では，$g=0.1$ の場合に一番速く，97.0 秒となった．これより，計算時間

表 5 各並列度で冗長化したネットワークを 50000 サイクル動作させた時の各経路に対応する状態の出現回数
Table 5 The numbers that the states corresponding to paths appeared when the model with a redundant representation was performed 50000 cycles using different g(placement Ⅰ).

経路の長さ	26.38	26.78	26.90	他の経路
g=0.05	7243	3365	1686	1259
g=0.25	6171	1455	647	3348
g=0.50	6068	2261	2051	2670
g=0.75	11947	4993	1411	3585
g=1.00	3613	1805	1071	1286

は約半分に減ることが確かめられた．

このような高速化は次の二つの理由によって達成されたと考えることができる．その一つは，単なる動作並列化からきた高速化である．もう一つは，Hopfield が示している探索空間の連続値化による効果と考えられる[2]．よく知られるように，素子の状態が 0 と 1 の 2 値しかとれない最初の Hopfield ネットワークにおいては，ネットワークの状態が超立方体の頂点から頂点へと遷移するが，素子を (0, 1) 上の連続値を取るようにすることで，ネットワークの状態は超立方体の内部をさ迷いながら問題の解に相当する頂点に近づくようになる．その結果，ネットワークはエネルギー関数の極小点に捕まりにくくなり，最小点を見つける可能性が高くなる．それは，上坂[12]に指摘されたように，内部の各点が実は頂点でのエネルギー関数値に関する情報を持っており，ネットワークはそれを頼りに動き，性質の良い頂点（解）に収束することができるためである．ボルツマンマシンの場合にも，似た効果を内部表現を冗長化することによって得ることができる．それは，冗長化したネットワークで複数の素子が同時に状態を更新するとき，もとの空間 $\boldsymbol{x}$ で各素子の状態が [0, 1] 上で微小変化することに相当するからである．それでも，ボルツマンマシンの確率的性質は保存され，エネルギーの小さい状態は高い確率で出現する．

配置 Ⅱ と配置 Ⅲ については，それぞれ $g=0.25$, 0.50, 0.75 の三つの場合に対して実験を行った．実験に用いられたパラメータ値と実験結果はそれぞれ**表 6** と**表 7** に示されている．初期状態を異なるものにして 50 回試行して得た結果である．配置 Ⅲ では，長さ 17.23 の最短経路に対して，長さ 17.27 の経路をはじめ，最短経路に非常に近い経路が多数ある．したがって，配置Ⅰより幾分難しい問題になっているが，平均

表 6 冗長化したネットワークで，配置Ⅱに対して，並列度で最短経路を見いだすまでの最大 (Max)，最小 (Min)，平均 (Mean) サイクル数
Table 6 The maximum, minimum, and mean numbers of cycle that the shortest path was found firstly when the model with a redundant representation was performed using different g(placement Ⅱ).

A	B	λ	g	Max	Min	Mean
3.0	5.0	0.05	0.25	575	38	207.0
3.0	6.0	0.06	0.50	388	35	145.2
3.0	7.0	0.07	0.75	1765	57	454.7

表 7 冗長化したネットワークで，配置Ⅲに対して，各並列度で最短経路を見いだすまでの最大 (Max)，最小 (Min)，平均 (Mean) サイクル数
Table 7 The maximum, minimum, and mean numbers of cycle that the shortest path was found firstly when the model with a redundant representation was performed using different g(placement Ⅲ).

A	B	λ	g	Max	Min	Mean
2.5	4.0	0.04	0.25	6264	113	1097.1
2.5	5.0	0.05	0.50	3559	51	883.8
2.5	6.3	0.06	0.75	6344	75	1387.2

1000 サイクル程度で最短経路を見つけ出すことに成功している．

実験結果をみると，必ずしも並列度が高くなるほど，最小点を早く見つけられるわけではないことが分かる．並列度が低すぎると並列動作の利点が得られなくなってしまう．並列度が高すぎると $|\Delta z_{ij}|$ が大きくなる可能性が出てきて，理論的にエネルギー最小状態への収束を保証できなくなる．どのような並列度で最も早く最小点が見つけ出せるのか，理論的には未だ分かっておらず，経験によって調整しなければならない．

脳の動作を考えると，脳細胞が一つずつ逐次的に状態を更新することはないが，すべての細胞が同時に状態を更新することもありえないので，脳の場合にも部分的並列化となっている．脳は非同期的な方式で動作すると考えられている．各脳細胞が独立に自分のクロックで状態更新を行う．ある細胞の状態更新がまだ終わっていないうちに，他の細胞は状態更新を始める可能性がある．ある時刻に着目したとき，状態更新を行っている素子の集団があるであろう．その集団を前章の状態更新のスケジューリングにおける集合 U_s と考えれば，本論文の手法は，脳の並列動作の自然なモデルになっている．ハードウェアの実現を考えても，このような非同期的な方式には，クロッキングスキーム (clocking scheme) が必要ではないので，より実現しやすいと考えられる．

7. おわりに

従来のボルツマンマシンでは，同時に複数の素子の状態を更新すると，熱平衡状態でエネルギー最小点の出現確率が最大となることが保証されていなかった．したがって，最適化問題等へ応用する場合にも，素子の状態を一つずつ逐次的に更新していかなければならず，並列計算を行うことができなかった．

本論文では，まず，従来のボルツマンマシンでは並列動作ができない原因について説明した．続いてそれを克服する方法として，内部表現の冗長化による動作並列化方法を提案し，それにより高度な並列動作を得られる可能性を直観的に示した．

後半では，巡回セールスマン問題を例として，具体的な冗長化方法と並列度を高める方法について説明し，また冗長化による動作並列化の効果をシミュレーションにより検証した．10 都市の巡回セールスマン問題に対するシミュレーションの結果によれば，提案した並列化手法を用いると，ボルツマンマシン中の全素子を同時に状態更新しても，熱平衡状態で，エネルギー最小状態の出現確率が最大となる性質が保たれることが確かめられた．

更に，表現を冗長化したために，素子数が増え，また非線形関数の計算が必要になる等のデメリットが生じるが，それを補い得る次のようなメリットが得られることが判明した．以下に示すのは，本文中に述べた 10 都市の巡回セールスマン問題を具体例としたデータである．

- 並列ハードウェアを用いる場合，冗長化により素子数は約 10 倍となり，また非線形関数計算のための特殊なハードウェアが必要となる．しかしながら，最適解を見いだすまでの時間（サイクル数）は，従来方式と比べ，約 1/50 に短縮される．
- 提案した並列化方式を逐次的計算機上でシミュレートする場合にも，従来方式より早く最適解を見いだすことができる．具体的には，論文中の条件の下で計算時間は約 2 分の 1 となった．

10 都市の巡回セールスマン問題に対する実験結果だけで，内部表現冗長化が常に有効であると結論することは危険である．今後の課題としては，本論文で提

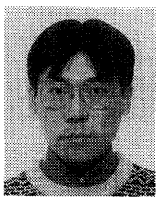
案した内部表現の冗長化によるボルツマンマシンの動作並列化方法に対し，より厳密な解析と大規模な検証実験を行う必要がある．

謝辞 日頃，御助言，御討論頂く本学小川英光教授，示唆に富む貴重な御意見を頂いた査読者の方に感謝いたします．

参考文献

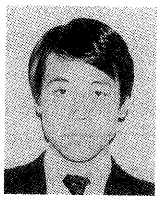
1) 麻生英樹：ニューラルネット情報処理，産業図書 (1988)．
2) Hopfield, J. J. and Tank, D. W.: Neural Computation of Dicisions in Optimation Problem, *Biol. Cybern.*, Vol. 52, No. 3, pp. 141-152 (1985)．
3) Hopfield, J. J.: Neural Networks and Physical System with Emergent Collective Computational Abilities, *Proc. Natl. Acad. Sci. USA*, Vol. 79, No. 8, pp. 2554-2558 (1982)．
4) Hinton, G. E., Sejnowski, T. J.: Optimal Perceptual Inference, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington D. C., pp. 448-453 (1983)．
5) Kirkpatrick, S., Gilatt, C. D. and Vecci, M. P.: Optimizing by Simulated Annealing, *Science*, Vol. 220, No. 4598, pp. 671-680 (1983)．
6) Aarts, E. and Korst, J.: *Simulated Annealing and Boltzmann Machines*, Wiley, Chichester, pp. 136-151 (1989)．
7) Takeda, M. and Goodman, J. W.: Neural Network for Computation: Number Representations and Programming Complexity, *Applied Optics*, Vol. 25, No. 18, pp. 3033-3046 (1986)．
8) Derthick, M. and Tebelskis, J.: 'Ensemble' Boltzmann Units Have Collective Computational Properties Like Those of Hopfield and Tank Neurons, *Neural Information Processing Systems* (ed. Anderson, D. Z.), American Institute of Physics, pp. 223-232 (1988)．
9) 熊沢逸夫：確率的ニューラルネットワークによる組合せ問題最適化における内部表現冗長化の効果について，信学論 (D-Ⅱ), Vol. J 72-D-Ⅱ, No. 10, pp. 1704-1712 (1989)．
10) 熊沢逸夫：内部表現を冗長化したニューラルネットによる巡回セールスマン問題の最適化，信学論 (D-Ⅱ), Vol. J 72-D-Ⅱ, No. 10, pp. 1713-1722 (1989)．
11) 太田 淳ほか：連想光ニューロコンピュータ，信学技報，**OQE 87-174** (1987)．
12) 上坂吉則：ニューロダイナミックスによる問題解決，ニューロコンピュータの現状と将来（甘利俊一監修，日本学際会議編）pp. 140-168, 共立出版 (1990)．

(平成4年8月24日受付)
(平成5年1月18日採録)



徐 丹

昭和38年生．昭和61年中国科学技術大学計算機科学卒業．平成5年東京工業大学大学院情報工学専攻博士課程修了．工学博士．現在，(株)サン・ジャパンに勤務．ニューラルネットワークの研究に従事．電子情報通信学会，神経回路学会各会員．



熊沢 逸夫（正会員）

1959年生．1981年東京工業大学工学部電気電子工学科卒業．1986年同大学院博士課程修了．同年同大工学部情報工学科助手．1990年同大学院博士課程修了．同年同大工学部情報工学科助手．1990年同学科助教授．パターン認識，信号画像処理，ニューラルネットワークの研究に従事．工学博士．電子情報通信学会，国際ニューラルネットワーク学会各会員．E-mail: kumazawa@cs.titech.ac.jp.