

遺伝的アルゴリズムの基礎と応用〔Ⅲ〕

小林 重信

前回は、GAの組合せ最適化問題への適用例として巡回セールスマン問題および看護婦スケジューリング問題を取りあげ、バランスのとれた探索(Well-Balanced Search)を実現するための設計上および実装上の創意工夫を紹介した。

今回は、当初、ジョブショップ型スケジューリング問題を取りあげる予定であったが、本誌本号には特集「遺伝的アルゴリズム」が組まれており、西川・玉置両氏による「遺伝的アルゴリズムと最適化」の中で、ジョブショップ型スケジューリング問題への適用が論じられているので、重複を避けるため、話題を変え、ニューラルネットワークへの適用を取りあげることにした。

7. ニューラルネットワークへの適用

7.1 問題の所在

Rumelhartら[1]によって提唱されたバックプロパゲーション型ニューラルネットワーク(BPN)は、これまでさまざまな分野に適用され、その有用性が実証されてきた。一般に、ニューラルネットワークの情報処理能力は、解の存在可能性および学習可能性の2つによって評価される。前者は解が存在するようなネットワーク構造を具体的に決定する問題であり、後者はネットワーク規模の拡大に伴う学習速度の低下や局所解の存在などを扱う問題である[2]。

Funahashi [3]は、“十分な隠れユニットの存在を許せば任意の連続写像を任意の精度で実現する3層BPNが存在する”(連続写像実現の完全性定理)ことを証明している。しかし、現実の問題に必要とされる隠れユニットの規模については、明らかにされていない。

また、ネットワーク規模の拡大に伴う学習速度の低下や局所解の問題は、非線形最適化問題に本質的に付随する問題でもあり、完全な学習を保証する解決方法は知ら

れていない。

ニューラルネットワークに関する決定問題は、次の2つからなる。

(1) 構造決定問題

入力層と出力層は先験的に与えられたとして隠れ層を何層とするか、各隠れ層におけるユニット数をいくつにするか、ユニット間の接続(結合構造)をどうするか、などが問題とされる。

(2) 重み決定問題

与えられた結合構造の下で、各結合の強さ(重み)をどのように設定するかが問題とされる。

BP学習則[1]は、フィードフォワード型結合構造の下で、誤差関数を極小化する重みの学習を保証している。学習の結果、0またはそれに近い重みをもつ結合を切り離すことにより、ネットワーク構造が獲得されたと結論づけるわけにはいかない。与えられたネットワーク構造には冗長な結合が存在していたと主張できるにすぎない。

与えられた問題に対し最適なニューラルネットワークとは、最適な構造と最適な重みを有するものでなければならないが、構造と重みを同時に最適化することは困難であることから、一般に、2レベルの決定問題として定式化し、上位レベルの構造決定問題は人間の直感的または経験的判断に頼り、下位レベルの重み決定問題のみをBP学習則などに委ねてきたのが実情である。

しかし、ニューラルネットワークによってより高次の情報処理を実現させることを考えた場合、構造決定問題を避けて通るわけにはいかない。ここに、構造決定問題に対する方法論の確立が要請される。

7.2 GAによる接近

前回は指摘したように、GAの特徴の1つは構造的表現を操作の対象としうる点にある。GAは、ニューラルネットワークの構造決定問題に対して、現時点で唯一有望な接近法と考えられる。GAによるニューラルネットワークの構造決定問題への接近は、歴史が浅く、数年前に開始されたばかりである[4]—[6]。

こばやし しげのぶ

東京工業大学 大学院総合理工学研究科知能科学専攻
〒227 横浜市緑区長津田4259

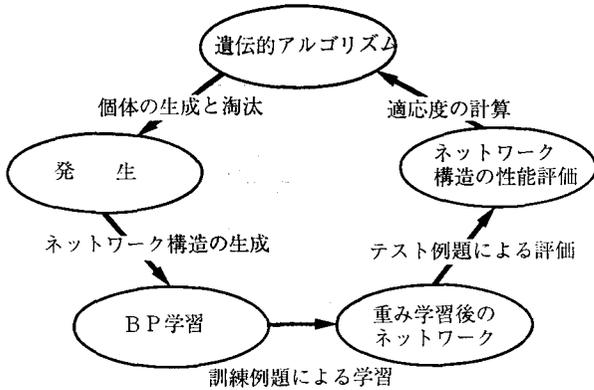


図1 GAによるネットワーク構造決定問題への接近

図1は、GAによる構造決定問題への接近の一般的な枠組みを示している。各個体はネットワーク構造をつくるための設計図に相当する染色体によって特徴づけられており、遺伝的操作によってつくられた新しい個体は、染色体に書かれている設計図に従って、発生過程により、ネットワーク構造を生成する。この段階では、ユニット間の接続関係だけが与えられ、結合の重みは考慮されない。結合の重みは、訓練例題集合を刺激として与え、たとえばBP学習則を使って決定する。重みが学習されたネットワークはテストフェーズに移され、性能が評価される。評価の結果にもとづいて適応度が計算され、GAに処理が移される。

ここで、ネットワーク構造をつくるための設計図をどのように定義するか、すなわちコード化が重要なポイントになる。

以下では、GAによる構造決定問題への接近のいくつかを紹介する。

7.3 直接コード化

ネットワーク構造を直接、コード化に反映させる方法を直接コード化 (Direct Encoding) という。

Miller and Todd [4] は、接続行列を使って、直接

from unit:	1	2	3	4	5	bias	
to unit	1	0	0	0	0	0	----->000000
	2	0	0	0	0	0	----->000000
	3	1	1	0	0	0	----->001101
	4	1	1	0	0	0	----->001101
	5	0	0	0	0	0	----->001101

000000 000000 001101 001101 001101

図2 直接コード化によるネットワーク構造の表現

コード化する方法を提案している。すなわち、最大許されるユニット数をNとするとき、

- ① 1列～N列：ユニットの出力リンク、
- ② 1行～N行：ユニットの入力リンク、
- ③ N+1列：ユニットのバイアス、

を表わす $N \times (N+1)$ の行列によって、ネットワーク構造を直接コード化するアイデアである。

図2に、入力層が2ユニット、隠れ層が2ユニット、出力層が1ユニットの場合のコード化の例を示す。この方法では、問題空間 (ネットワーク構造) とGA空間 (遺伝子型表現) が1対1に対応しており、3節で議論したコード化の評価規範は満たされている。

この方法では、染色体が接続行列で表現されることから、交叉オペレータとしては、2つの親の間で、

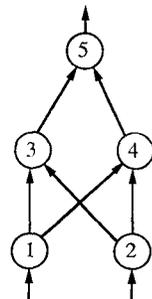
- ① 行または列の交換、
- ② 部分行列の交換、

などが考えられる。

また、突然変異オペレータとしては行列の任意の要素の値を反転させることが考えられる。図3は、交叉オペレータによって2つのネットワーク構造の組替えにより、新しいネットワーク構造が生成される様子を示している。

Miller and Todd [4] は、本方法の有効性を確認するために、XOR問題、4象限問題、パターン複写問題での実験結果を報告している。

まず、XOR (排他的論理和) 問題への適用結果であるが、図4の (a) および (b) は、4ユニットまたは5ユニットを用いた場合の人間が考える標準的なネットワーク構造であるのに対し、(c) は、GAによって見いだされた非対称なネットワーク構造である。(c) では、入力リンクから出力リンクへの直接結合が重みの調整を早め、XOR問題の学習が効率的なものになることが理解され



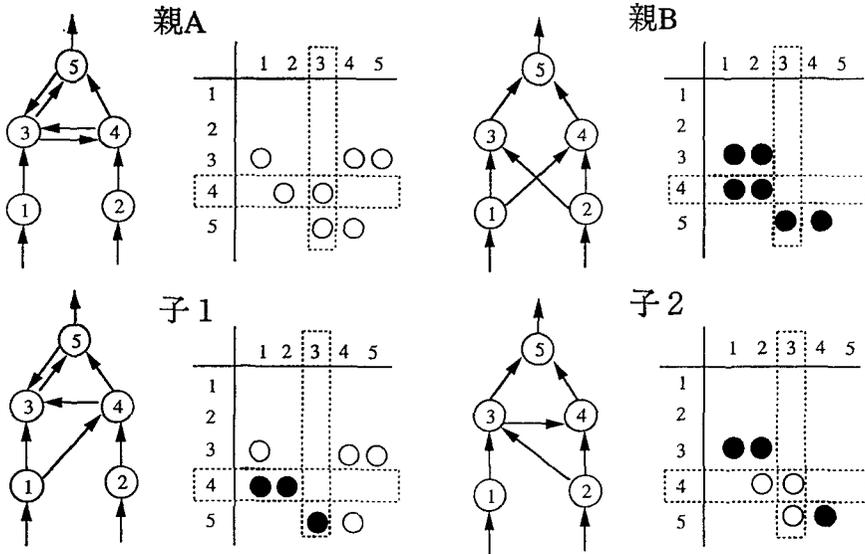


図 3 交叉によるネットワーク構造の組み換え

る。

4象限(Four-Quadrant)問題とは、図5の(a)に示されるもので、区間 $[0, 1]$ の値をとる2つの入力変数の組によって4分割された2次元座標に対して、入力点がどの象限に属するかによって0または1を出力する問題である。この問題に対し、人間が与えるネットワーク構造は(b)または(c)であるのに対し、GAが見いだしたネットワーク構造は(d)であり、人間の直感を越えたネットワーク構造が見いだされるところに本法の特徴があるとしている。

しかし、XOR問題および4象限問題は問題の規模が小さくて、いずれも1世代で最適解が見つかり、10世代で収束しており、GAの有効性を主張することは躊躇される。

パターン複写 (Pattern Copying) 問題とは、入力パターンを出力パターンとしてそのまま出力す

る問題をいう。隠れ層を使わずに、10個の入力ユニットを、直接、10個の出力ユニットに連結するものとする。この問題に対する可能なネットワーク構造は 2^{100} 通りあ

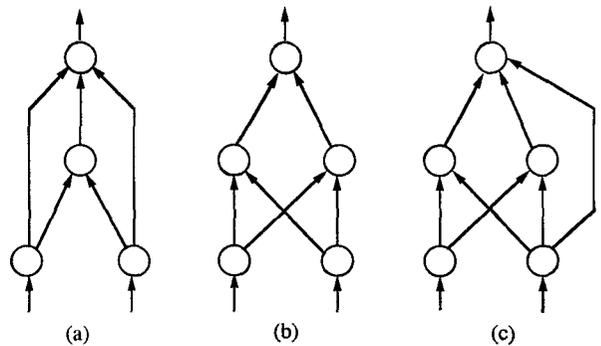


図 4 XOR問題のネットワーク構造

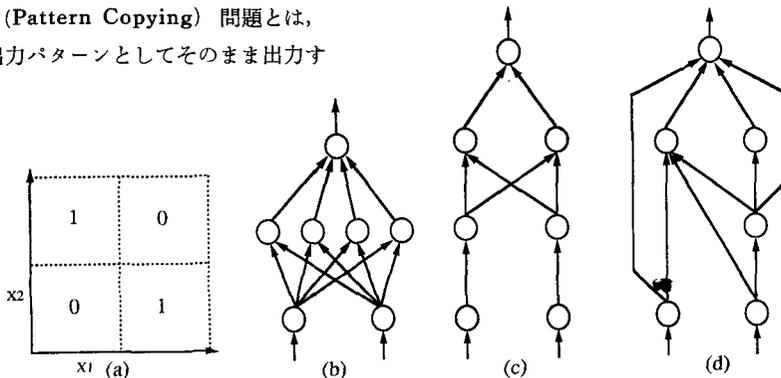


図 5 4象限問題のネットワーク構造

る。GAを使って探索させた結果、約6世代で正しいネットワーク構造が見つかり、15世代でほとんど収束したと報告されている。ちなみに、突然変異だけで探索させたところ、20世代目で正しいネットワーク構造が見いだされたと報告している。

Miller and Toddの方法は、簡潔かつ明快な枠組みを有するが、ネットワーク構造を陽に指定する、という意味において、強い指定方法 (strong specification method) であり、対象とするネットワーク構造の規模が大きくなるにつれて、計算量の問題が顕在化することが予想される。

Harpら[5]は、直接法の問題を克服する試みとして、ネットワーク構造を詳細に指定することはないで、ネットワーク構造の骨組みだけを指定する方法を採用している。すなわち、あらかじめネットワーク構造をいくつかの部分構造に分割し、部分構造内および部分構造間でのユニットの結合を、それぞれ弱く指定する (weakly specify) もので、たとえば“部分構造Aと部分構造Bの間には5本のリンクで結合される”というような形で指定する。指定の仕方に対象領域の知識を反映させることができるという点で、優れた方法といえるが、実際の効果は不明である。

指定の方法の強弱の違いはあれ、直接法にもとづいて大規模なネットワーク構造を探索することは、計算量的な観点からみて効率的是とはいえない。

7.4 間接コード化

インタプリタを介して、環境を考慮した上で、コード化された内容を翻訳し、ネットワーク構造を生成するようなコード化方法を間接コード化 (Indirect Encoding) という。

図6に、直接コード化と間接コード化の違いを示す。直接コード化では、染色体とネットワーク構造が1対1に対応する。間接コード化では、同じ染色体であっても、環境が異なれば、異なったネットワーク構造を生成する可能性がある。また、環境が同じであっても、染色体とネットワーク構造の対応関係は1対1ではなく、一般に、多対1となる。環境が異なれば、染色体とネットワーク構造の対応関係は多対多となり、非決定的となる。

間接コード化の1つとして文法コード化 (Grammatical Encoding) がある。文法コード化とは、グラフ

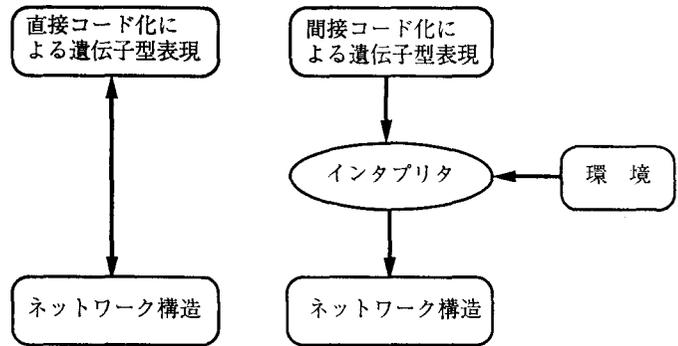


図6 直接コード化と間接コード化

生成規則の集まりによって染色体を特徴づけ、発生過程を通じてネットワーク構造を生成する方法をいう。

Kitano[6]は、L-システム[7]を拡張したグラフ生成文法を用いてネットワーク構造の生成を行なう方法を提案している。

たとえば、ある個体Xの染色体が次のようなグラフ生成規則の集まりによって特徴づけられているとする。

$$\begin{aligned}
 S &\rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} & A &\rightarrow \begin{bmatrix} d & e \\ a & f \end{bmatrix} & B &\rightarrow \begin{bmatrix} a & a \\ c & a \end{bmatrix} \\
 C &\rightarrow \begin{bmatrix} a & b \\ a & a \end{bmatrix} & D &\rightarrow \begin{bmatrix} b & a \\ a & a \end{bmatrix} & a &\rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
 b &\rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} & c &\rightarrow \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} & d &\rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
 e &\rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & f &\rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & g &\rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}
 \end{aligned}$$

初期状態Sに対して、これらの生成規則を適用すると、次のような接続行列が得られる。

$$\begin{aligned}
 S &\rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} \rightarrow \begin{bmatrix} d & e & a & a \\ a & f & c & a \\ a & b & b & a \\ a & a & a & a \end{bmatrix} \\
 &\rightarrow \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

同様に、別の個体Yの染色体が次のようなグラフ生成規則の集まりによって特徴づけられているとする。

$$\begin{aligned}
 S &\rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} & A &\rightarrow \begin{bmatrix} b & a \\ a & b \end{bmatrix} & B &\rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\
 C &\rightarrow \begin{bmatrix} a & g \\ a & a \end{bmatrix} & D &\rightarrow \begin{bmatrix} e & a \\ a & e \end{bmatrix} & a &\rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\
 b &\rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} & c &\rightarrow \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} & d &\rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
 e &\rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & f &\rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} & g &\rightarrow \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}
 \end{aligned}$$

ここで、個体Xと個体Yでは、A~Dの書き換え規則だけが異なっている。個体Yに対して、初期状態Sとして、これらの生成規則を適用すると、次のような接続行列が得られる。

$$\begin{aligned}
 S &\rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} \rightarrow \begin{pmatrix} b & a & a & b \\ a & b & c & d \\ a & g & e & a \\ a & a & a & e \end{pmatrix} \\
 &\rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

個体Xと個体Yの間での交叉によって、AとBの書き換え規則は個体Xより、CとDの書き換え規則は個体Yより受け継いだ子Zが生成されたとする。Zは次のような書き換え規則によって特徴づけられる。

$$\begin{aligned}
 S &\rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} & A &\rightarrow \begin{bmatrix} b & a \\ a & b \end{bmatrix} & B &\rightarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\
 C &\rightarrow \begin{bmatrix} a & g \\ a & a \end{bmatrix} & D &\rightarrow \begin{bmatrix} e & a \\ a & e \end{bmatrix} & a \sim g &\text{はX, Y} \\
 & & & & &\text{と同じ}
 \end{aligned}$$

個体Zの生成規則によってつくられる接続行列は次のようになる。

$$S \rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} \rightarrow \begin{pmatrix} d & e & a & a \\ a & f & c & a \\ a & g & e & a \\ a & a & a & e \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Kitano[6]は、グラフ生成規則を使った間接コード化の有効性を確認するために、8×8エンコーダ・デコーダ問題を例に、直接コード化との比較実験を行なっている。

エンコーダ・デコーダ問題とは、パターン複写問題の特殊なもので、入力パターンの1つの要素のみが1で、他は0であり、入力パターンと同じものを出力パターンとして生成する問題をいう。

32×32の接続行列を2つの方法を用いて獲得させたところ、個体数が10と100の場合について、いずれの場合もオフライン性能(収束性)およびオンライン性能(収束速度)の両方において、間接コード化の方が、直接コード化に比べて、優れた性能を示したと報告している。

Kitano[6]によれば、2つの方法で得られた接続行列を比較すると、直接コード化による接続パターンには規則性が観察されないのに対して、間接コード化による接続パターンには局所的に強い結合をもつ規則的な部分構造が存在していることが観察され、その理由は生成規則の繰返しまたは再帰的な適用の効果に求められるとしている。

規則的な接続パターンを生成することができる間接コード化は、たとえば音声信号処理のように時間遅れを伴うパターン情報を処理する際に威力を発揮するものと期待される。

7.5 考察

B P型ニューラルネットワークは、パターン情報処理の有力手段として広く使われている。しかし、ネットワーク構造の同定については、これまで試行錯誤以外の方法をもたなかった。GAは、ネットワーク構造の決定問題に対する有力な接近法と思われる。

また、生成規則を使って構造を生成するアイデアは生物の発生過程からの着想でもあり、進化と発生を融合した枠組みは大規模システムの設計原理としても非常に魅力があり、今後の発展が期待されることである。

今回は、ニューラルネットワークの構造決定問題への適用を通じて、直接コード化と間接コード化を中心に解説した。GAは構造的表現を直接扱えるだけでなく、

間接コード化を導入することにより、より大規模な問題に対し、GAの適用の道が開かれていることを示唆した。

今回は、最終回であり、理論的な話題と学習の問題に言及する予定である。

参 考 文 献

- [1] Rumelhart, D. E. and McClelland, J. : Parallel Distributed Processing : Explorations in the Microstructure of Cognition, I, II and III, MIT Press (1986, 1987).
- [2] Funahashi, K.: On the Approximate Realization of Continuous Mapping by Neural Networks for Invariant Pattern Recognition, Neural Networks. Vol. 2, pp.183—192 (1989).
- [3] 田中, 山村, 小林: 高次バックプロパゲーションネットワークの情報処理能力に関する考察, 計測自動制御学会論文集, Vol. 28, No. 1, pp. 125—134 (1992).
- [4] Miller, G., Todd, P. and Hegde, S. : Designing Neural Networks using Genetic Algorithms, Proc. of 3-rd ICGA, pp. 379—384 (1989).
- [5] Harp, S. A., Samad, T. and Guha, A. : Towards the Genetic Synthesis of Neural Networks, Proc. of 3-rd ICGA, pp. 360—369 (1989).
- [6] Kitano, H. : Designing Neural Networks using Genetic Algorithms with Graph Generation Systems, Complex Systems, Vol.4, No.4 (1990).
- [7] Lindenmayer, A. : Mathematical Models for Cellular Interactions in Development, J. Theor. Biol., Vol. 18, pp. 280—299 (1968).

●最新刊●

Perlの国へようこそ

前田 薫他著

A5・定価2472円

ユーザのためのUNIX

鬼頭繁治著

A5・定価2781円

ツールとしてのUNIX

深澤良彰編著

A5・定価2060円

現代数学の風景

野崎昭弘編著

A5・定価2781円

数学のための英語案内

野水克己著

A5・定価2000円

新時代のコンピュータ総合誌

Computer Today

7月号/発売中/定価930円

特集

ステレオグラム進化論

3Dステレオグラムはなぜ立体に見えるのか? どうやったら作れるのか? そのしくみとプログラムを公開。

3次元を復元する

2次元図を読みこんで3次元図形を構築するための研究と製品を紹介する。

月刊誌

数理科学

7月号/発売中/定価930円

可逆と非可逆

可逆と非可逆を巡って/物理学における時間の矢/量子コンピュータにおける可逆と非可逆/経済現象における可逆と非可逆/物理現象における可逆と非可逆

サイエンス社

東京都千代田区神田須田町2-4 安部徳ビル

☎ 03-3256-1091 振替 東京7-2387