実数を遺伝子とした遺伝的アルゴリズムによるデータあてはめ

吉本 富士市[†] 原田 利宣[††]
森山 真光[†] 吉本 芳英[†††]

スプラインを用いたデータあてはめ問題では，良い近似関数を得るためには，節点を変数として扱う必要があることが多い．そのとき，解くべき問題は多変数で多峰性の連続系非線形最適化問題となる．したがって，その大域的な最適解を求めることは困難である．本論文では，実数を遺伝子とした遺伝的アルゴリズムを用いて，この問題を解く方法を提案する．この方法は，節点をそのまま遺伝子とするので，元の連続系の問題を離散系の組合せ問題に変換する必要がない．このため，節点の離散化による誤差の影響を避けることができ，準多重節点を作ることも可能である．あてはめの評価関数として，情報量規準 BIC（Bayes Information Criterion）を用いて最適なモデルを探索する．節点は，あらかじめ良い初期値を設定しなくても，その適切な数と位置を，自動的かつ同時に求めることができる．また，ユーザが主観的な判断によって決めるパラメータは必要でない．さらに，準多重節点を多重化する簡単なアルゴリズムを提案する．本論文の方法は，データの元にある関数（underlying function of data）がなめらかなデータはむろんのこと，不連続なところや尖ったところがあるデータも扱うことができる．この方法の有効性を示すため，数値計算例をあげている．

Data Fitting by a Genetic Algorithm with Real Number Genes

Fujiichi Yoshimoto,[†] Toshinobu Harada,[††]
Masamitsu Moriyama[†] and Yoshihide Yoshimoto[†††]

In order to obtain a good approximation for data fitting with a spline, frequently we have to deal with knots as variables. Then, the problem to be solved becomes a continuous nonlinear and multivariate optimization problem with many local optima. Therefore, it is difficult to obtain the global optimum. In this paper, we propose a new method for solving the problem by a genetic algorithm with real number genes. In this method, we use knots themselves as genes, and we do not convert the original continuous problem into a discrete combinatorial problem. Therefore, influence of the errors caused by the discretization of knots is avoided, and quasi-multiple knots can be constructed. We search for the best model among candidate models by using Bayes information criterion, BIC. Our method can determine appropriate number and locations of knots automatically and simultaneously without good initial locations of knots. We do not need any subjective parameters. Moreover, we propose a simple algorithm for multiplying quasi-multiple knots. Our method can treat data not only with a smooth underlying function but also with an underlying function having discontinuous points and/or cusps. Numerical examples are given to show the performance of our method.

1. まえがき

スプラインを用いたデータあてはめは，実験データの処理，形状モデリングなどの重要な要素技術の 1 つである．よく知られているように，良いスプライン（良いモデル）を得るためには，通常は節点の数と位置を適切に決める必要がある．このとき，節点を変数として扱わなければならず，解くべき問題は多変数で多峰性の連続系非線形最適化問題となる[1),2)]．したがって，大域的な最適解を求めることはきわめて困難である．

このため，上記の最適化問題をまともに解かない方法（簡便法）がいろいろ提案されてきている[1)~9)]．しかし，これらの方法は，許容誤差または平滑化係数（smoothing factor）が必要であり節点の数も多い[2),7)]，節点の数が多い[3)]，適切な初期節点の配置が容易でな

† 和歌山大学システム工学部情報通信システム学科
Department of Computer and Communication Sciences, Faculty of Systems Engineering, Wakayama University
†† 和歌山大学システム工学部デザイン情報学科
Department of Design and Information Sciences, Faculty of Systems Engineering, Wakayama University
††† 東京大学大学院理学系研究科物理学専攻
Department of Physics, Graduate School of Science, University of Tokyo

い[6]，など改善の余地があるものが多い．したがって，“自動的に良いモデルを得る手法” の観点から見るとまだ十分とはいえない．

ここで “良いモデル” とは，データの元にある関数（underlying function of data）をできるだけよく近似し，しかもモデルのパラメータができるだけ少ないスプラインのことを意味する．自動的に良いモデルを得るためには，モデルの良さを評価するための客観的な規準を用いて適切な節点の数と位置を自動的に決めるアルゴリズムが必要である．しかし，そのための汎用性の高いアルゴリズムはまだほとんど提案されていない．

最近，吉本ら[10),11)]は，元の連続系の最適化問題を離散系の組合せ最適化問題へ変換し，それを遺伝的アルゴリズム（GA）を用いて解く方法を提案している．この方法は，離散化にともなう節点位置の誤差が避けられない．したがって，節点位置を高精度に求めたいときには遺伝子長を非常に長くする必要があり，計算量が多くなる．また，多重節点[12)]を扱うことができないため，構成されるモデル関数はなめらかなものに限定される．

この問題を解決するため，本論文では実数を遺伝子としたGAを用いて，データあてはめ問題を解く方法を提案する．この方法によれば，節点をそのまま遺伝子とすることができ，元の連続系の問題を離散系の組合せ問題に変換する必要がない．このため，節点位置の離散化による誤差の影響を避けることができ，準多重節点（3.6 節参照）が可能となる．また準多重節点は，5 章で述べる簡単なアルゴリズムにより多重節点化することができる．このため，データの元にある関数がなめらかなデータはむろんのこと，それが不連続なところや尖ったところを持っているデータも扱うことができる．

また，そのような特異点を持つデータを扱うためには，あてはめの評価関数として情報量規準BIC（Bayes Information Criterion）[13),14)]が適していることを示す．BICを用いても，AIC[14),15)]を用いた場合と同様に，節点の適切な数と位置を自動的かつ同時に求めることができる．提案する方法は，許容誤差とか平滑化パラメータなど，ユーザの主観的な判断に任されるものは必要としない．また，GAで最適解を探索するとき，節点の良い初期値も不要である．提案する方法の有効性を示すため，数値計算例をあげる．

2. スプラインによるデータあてはめ

簡単のため，ここでは要点だけを述べるので，詳しくは文献 1)，10) を参照されたい．あてはめを行うべきデータは，x 軸上の区間 $[a,b]$ 内で与えられ，

$$F_j = f(x_j) + \epsilon_j, \quad (j = 1, 2, \cdots, N) \tag{1}$$

と表されるものとする．ここで，$f(x)$ はデータの元にある関数であり，ϵ_j は平均値 0，分散 σ^2 の互いに独立な誤差であると仮定する．もちろん，$f(x)$ は未知の関数であり，その良い近似関数を作ることがデータあてはめの目的である．

必要な節点を $\xi_i\,(i = 1-m, 2-m, \cdots, n+m)$ と書くことにする．ここで，n は区間 $[a,b]$ の内部に配置する節点（内部節点）の数である．また m は，式 (3) に示すように，近似関数 $S(x)$ を表すために使うB-スプライン $N_{m,i}(x)$ の階数（次数+1）である．両端の m 個の節点はそれぞれ端点 a，b に重ね，

$$\left.\begin{aligned} a &= \xi_{1-m} = \cdots = \xi_0 \\ b &= \xi_{n+1} = \cdots = \xi_{n+m} \end{aligned}\right\} \tag{2}$$

とする．

このとき，近似関数 $S(x)$ は

$$S(x) = \sum_{i=1}^{n+m} c_i N_{m,i}(x) \tag{3}$$

と表すことができる[1)]．

式 (3) に含まれるB-スプラインは，次の漸化式を用いて容易に計算できる[16)]．

$$N_{1,i}(x) = \begin{cases} 1 & (\xi_{i-1} \le x < \xi_i), \\ 0 & (\text{otherwise}), \end{cases} \tag{4}$$

$$\begin{aligned} &N_{r,i}(x) \\ &= \frac{(x-\xi_{i-r})N_{r-1,i-1}(x)}{\xi_{i-1}-\xi_{i-r}} + \frac{(\xi_i - x)N_{r-1,i}(x)}{\xi_i - \xi_{i-r+1}}, \\ &\qquad (r = 2, 3, \cdots, m). \end{aligned} \tag{5}$$

式 (5) で，節点が分子および分母の両方に入っていることに注意したい．すなわち，節点はB-スプライン $N_{m,i}(x)$ の非線形パラメータである．したがって，式 (3) で与えられるスプライン $S(x)$ は節点の非線形関数である．また，式 (5) は節点を多重化した場合にもそのまま有効である[1)]．ただし，$0/0 = 0$ とする．

最小二乗法を用いて式 (3) を与えられたデータ (1) にあてはめるとき，残差の 2 乗和 Q は

$$Q = \sum_{j=1}^{N} w_j \{S(x_j) - F_j\}^2 \tag{6}$$

となる．ここで，w_j はデータの重みであり，$N > n+m$ とする．式 (6) を最小にする条件からB-スプライン係数 $c_i\,(i = 1, 2, \cdots, n+m)$ を求めることができる．ただし，良い近似を得るためには内部節点 $\xi_i\,(i = 1, 2, \cdots, n)$

の数と位置を適切に決める必要がある．そのためには節点を変数として扱わなければならないが，そのとき式 (6) を最小化する問題は多峰性の最適化問題となる[2),6)]．

3. 遺伝的アルゴリズムの適用

スプラインを用いたデータあてはめ問題に GA を用いる理由は，文献 10) を参照されたい．ここでは，初期個体と初期集団，評価関数，選択・交叉・突然変異の方法，準多重節点および制御パラメータの自動的な決定について述べる．

3.1 初期個体と初期集団

まず初期個体の生成方法を述べる．今，遺伝子長を L と書くことにする．その値は内部節点の数に等しく，初期個体生成時には $L = [\lambda N]$ と書くことができる．ここで，λ は節点率[10)]であり $0 \leq \lambda < 0.5$ とする．この下限値 0 は，内部節点がないときに対応する．また，上限値 0.5 は内部節点の数がデータ数 N の約 1/2 であることに相当する．この値は，あてはめの数値実験で計算が収束する上限から決めたものである．

λ を変化させることによって，内部節点の数の初期値を制御できる．その値は，$0 \leq \lambda < 0.5$ の中で自由に決めてよい．ただし，収束後の節点数になるべく近い値を用いた方が，計算量を少なくできる可能性が高い．なお，この値は各個体ごとに異なっていてもかまわないが，数値実験の結果によれば，そのようにしても特に利点はなかった．したがって，本論文では簡単のため一定の値とする．

図 1 に例を示すように，初期個体は，あてはめを行う区間 $[a, b]$ で一様な乱数を遺伝子長の数だけ（L 個）発生させ，それらを上昇順に並べ替えたもとする．初期集団は，その個体をあらかじめ決められた数（K 個とする）だけ作成したものである．ここで K は偶数である．この個体に含まれる遺伝子は実数であるが，それをそのまま初期節点とする．すなわち，コード化を行わないで個体を生成する．このようにすると，文献 10)，11) で生じた節点の離散化にともなう誤差は生じないことになる．

以上のように作成された個体を解候補の初期集団として，2 章で述べたあてはめの問題の最適解を大域的に探索する．図 1 のとおり，初期集団の遺伝子は各個体ごとに異なっているが，GA の計算が収束するにつれて同一のものが多くなっていく．すなわち，ランダムに配置された初期節点が，データの元にある関数に応じて，適切な数と位置の節点に収束していく．

遺伝子は，通常の GA ではビットであるので軽いが，

	gene 1	. . . . .			gene L
Individual 1	0.11	0.35	0.62	0.75	0.91
Individual 2	0.33	0.41	0.56	0.68	0.82
Individual 3	0.21	0.52	0.73	0.85	0.93
. . .		. . . . .			
Individual K	0.05	0.25	0.48	0.72	0.87

note 1: interval of fitting [a, b]=[0, 1].
note 2: significant digit is 2.

図 1 初期個体と初期集団の例
Fig. 1 An example of initial individuals and population.

上で述べた方法では実数であるので重たくなる．ただし，文献 10) で述べたコード化の場合に比べると，個体の表現に節点だけを用いているため，遺伝子長が短くなる利点がある．いずれにせよ，これは小さな問題である．なぜなら，6 章の例題 1 で示すように，通常は遺伝的操作（選択，交叉および突然変異）に要する時間は全体の計算時間に対して無視できるほど小さいからである．

3.2 評価関数

本論文では，あてはめの評価関数として情報量規準 BIC[13),14)]を用いる．この規準は，2 章で述べたあてはめの場合

$$BIC = N \log_e Q + (\log_e N)(2n + m) \qquad (7)$$

と表現できる．ここで，N はデータの数，Q は式 (6) で表される残差の 2 乗和である．さらに，$(2n + m)$ はモデル関数（近似関数）に含まれるパラメータの数である．この中で，$n + m$ は B-スプライン係数の数，n は内部節点の数である．

評価関数 BIC の値を評価値と呼び，それを最小にするモデルが最も良いモデルであると見なされる．AIC[15)]と同様に，BIC を用いる場合にも，従来の方法[2),7)]で必要とされる許容誤差とか平滑化パラメータは不必要となる．これらの適切な値を設定するためには，経験と試行錯誤を必要とする場合が多いので，BIC を用いる効果は大きい．

ところで，文献 10)，11) で AIC を用いたにもかかわらず本論文で BIC を用いることを提案する理由は，多くの例題で数値実験を行った結果による．データの元にある関数が大域的になめらかな場合には，AIC を用いた結果も BIC を用いた結果も大きな違いはなかったが，それに特異性がある場合には BIC の方がパラメータ数（節点数）の少ないモデル関数を得ることができた．

AIC を用いた場合には，BIC を用いた場合に比べて，一般に節点の数が多くなる傾向がある．このことは文献 14) の記述「AIC は少し大き目のモデルを選択する傾向がある」と一致する．特に，データの元にある関数が不連続なところや尖ったところを持っている場合には，AIC を用いると節点が不必要と思われるところにも入ることが多かった．6 章の例題で示すように，BIC を用いる場合にはそのようなことがなく，節点を適切に配置することができた．

なお，GA の文献[17)~20)]では，“適応度が大きいほど最適値に近い” と表現されている場合が多いが，本論文では BIC をそのまま評価関数として用いているため，“評価値が小さいほど最適値に近い” ことになるので注意されたい．また，BIC はデータ圧縮への応用でよく知られている MDL（Minimum Description Length）原理[21),22)]と本質的には同じである[14)]．

3.3 選択の方法

次世代の個体候補の選択には，文献 10) と同様にトーナメント方式を用いる．具体的には，個体集団の中からランダムに 2 個体抽出し，その中で評価値の良い方を次世代の個体候補とする作業を，K 回（個体数）だけ反復する．ただし，多様性を維持するため同じ個体は 3 度以上選択しない．

3.4 交叉の方法

文献 10) と同様に 2 点交叉を用いる．ところが，3.1 節で述べたように節点をそのまま遺伝子とすると，初期個体を除いて遺伝子長は一般に各個体ごとに異なる．したがって，GA で通常行われているように[17),20)]交叉させたい個体を同じ遺伝子座で切って間の遺伝子列を交換することはできない．そこで，あてはめの区間 $[a, b]$ をランダムに 2 点で切って，その間の遺伝子を交換する方式を用いる．このとき，交換する遺伝子の数は同じでないことが多いが，それは何ら問題を生じない．

このことを例で説明する．**図 2** は，あてはめの区間が $[a, b] = [0, 1]$ のとき，個体 A と B を交叉させる例である．図 2 (a) は交叉前の状態を示している．今，2 つの切断点が 0.5 と 0.7 になったと仮定する．このとき，個体 A の 0.64 と個体 B の 0.54 および 0.69 を交換することになる．したがって，交叉後は図 2 (b) に示すようになる．よって，個体 A の遺伝子長が 5 から 6 に，個体 B の遺伝子長が 6 から 5 に変更される．このような交叉を行うと，GA の計算が進むにつれて遺伝子長（内部節点の数）は動的に変化しながら最適な値へと収束していく．

1 世代全体のアルゴリズムは以下のとおりである．

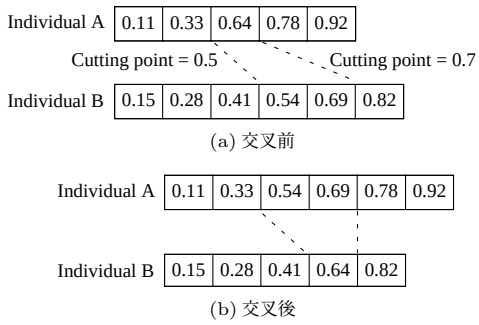


図 2 2 点交叉の例
Fig. 2 An example of two-point crossover.

個体 1 から個体 K までを，1 番から順番に 2 つずつ取り出しながら（すなわち，$i = 1, 3, 5, \cdots, K-1$ と変えながら）次の処理を行う．今，取り出した個体を個体 i および個体 $i+1$ とする．

ステップ 1： 個体 i および個体 $i+1$ に対して，3.7 節の式 (8)，(9) を用いて交叉率 p_c を計算する．

ステップ 2： 区間 [0, 1] で一様な乱数を発生させる．その値が p_c よりも大きければ，図 2 に示す交叉を行う．そうでなければ交叉を行わない．

3.5 突然変異の方法

節点をそのまま遺伝子として用いると，GA で通常行われているように[17),20)]個々の遺伝子を突然変異させる（ビット反転させる）ことはできない．そこで代わりに，遺伝子（節点）をある確率で追加または削除する方式を用いる．ただし，追加と削除の割合は均等にする．そのアルゴリズムは以下のとおりである．

各個体ごとに，反復回数のカウンタを 0 にした後，以下の操作を行う．

ステップ 1： 区間 [0,1] で一様な乱数を 1 つ発生する．その値が突然変異率 p_m 以下であればステップ 2 へ行く．そうでなければステップ 5 へ行く．

ステップ 2： 区間 [0,1] で一様な乱数を 1 つ発生する．その値が 0.5 以下であればステップ 3 へ行く．そうでなければステップ 4 へ行く．

ステップ 3： 区間 $[a, b]$ で一様な乱数を 1 つだけ発生し，その点に遺伝子（節点）を追加してステップ 5 へ行く．

ステップ 4： その個体に含まれる遺伝子（節点）の中から，ランダムに 1 つを選び削除してステップ 5 へ行く．

ステップ 5： 反復回数のカウンタを 1 増やす．その値が遺伝子長を超えていれば終了する．そうでなければステップ 1 へ戻る．

なお，上記のアルゴリズムを適用すると，遺伝子長

が動的に変更される．したがって，ステップ 5 の反復回数の終了判定に用いる遺伝子長は，突然変異を行う前の長さとする．

3.6 準多重節点

本章で以上述べた方法では，“真の意味での多重節点” を生成することは，残念ながらほとんど不可能である．なぜなら，乱数で同じ値を生成する確率はほとんどゼロであるからである．しかし，6 章の例題で示すように，“多重節点に近いもの” は生成できる．それらの結果から判断すると，厳密な意味での多重節点でなくても不連続なところや尖ったところのあるモデル関数に十分近いものを作ることができる．本論文では，そのような節点を “準多重節点” と呼ぶ．

もしも真の意味での多重節点が必要な場合には，本論文で提案する遺伝的アルゴリズム（4 章参照）で得られた結果を初期値として別の最適化アルゴリズムを適用し，準多重節点を多重節点化すればよい．5 章で，そのようなアルゴリズムの 1 つを提案する．

3.7 制御パラメータの決定

3.3 節で述べたように，個体の “選択” は親の個体数と同じだけランダムに選ぶ方式とするので，その確率を決める必要はない．しかし，“交叉” および “突然変異” については，それらの確率をどう決めるかが重要である．GA ではこれらの値は固定されている場合が多いが，本論文では Srinivas らの方法[23]を適用して適応的に決めることにする．その概要は以下のとおりである．

今，ある世代で交叉させる 2 つの個体の交叉率を p_c と書くことにすると，

$$p_c = (e' - e_{\min})/(\bar{e} - e_{\min}), \quad e' \le \bar{e}, \tag{8}$$

$$p_c = 1.0, \quad e' > \bar{e} \tag{9}$$

となる．また，その世代のある個体の突然変異率を p_m と書くことにすると

$$p_m = \max\left\{\frac{0.5(e - e_{\min})}{\bar{e} - e_{\min}}, P_{c,\min}\right\}, \quad e \le \bar{e}, \tag{10}$$

$$p_m = 0.5, \quad e > \bar{e} \tag{11}$$

となる．ここで，e はある個体の評価値，$e_{\min}$ は K 個の集団の中で最小の評価値，$\bar{e}$ は K 個の集団の評価値の平均値，e' は交叉させる 2 つの個体の中で評価値の小さい方の値である．また，$P_{c,\min}$ は最小突然変異率を意味する．3.2 節で述べたように，本論文では評価値が小さいほど良い解であるので，文献 23) の $f_{\max}$ が本論文では $e_{\min}$ に変更されているなど，表現にいくつかの変更があるので注意されたい．

式 (8)，(9) のようにすると，$\bar{e}$ 以上の評価値を持つすべての個体は必ず交叉させることになる．また，交叉の確率は e' が $e_{\min}$ に近づくにつれて小さくなり，$e_{\min}$ に等しい評価値を持つ個体に対しては 0.0 となる．さらに，式 (10)，(11) のようにすると，$\bar{e}$ 以上の評価値を持つすべての個体は突然変異率が 50%となるので，完全に破壊されることになる．また，突然変異率は e が $e_{\min}$ に近づくにつれて小さくなるが，最小突然変異率 $P_{c,\min}$ よりも小さくはならない．このようにする理由は，もしも p_c と p_m がともにゼロとなれば，その個体はそのまま次の世代へと引き継がれることになり，初期収束を起こす可能性が高くなるからである[23]．このため，最小突然変異率 $P_{c,\min}$ をあらかじめ設定しておく．$P_{c,\min}$ の値は，たとえば 0.01 である．

なお，突然変異率を式 (10) のようにすると，すべての最良個体が破壊されてしまう可能性があるので，出現した個体の中で最良のものを記録しておく必要がある．

4. GA を用いたデータあてはめのアルゴリズム

3 章で述べた “実数を遺伝子とする GA” を用いたデータあてはめのアルゴリズムは，次のようになる．大まかに見ると文献 10)，11) の方法と似ているが，3 章で述べたとおり初期集団の作り方と遺伝的操作（交叉，突然変異）の内容はまったく異なる．また，ユーザが与える制御パラメータの数が少なくなっており，いっそうの自動化が行われている．3.1 節で述べたとおり，GA による最適化計算の初期値（初期個体）は乱数を用いて決めているので，それをユーザが設定する必要はない．

ステップ 1： あてはめの区間 $[a, b]$ と，式 (1) で表される，あてはめを行うべきデータを入力する．

ステップ 2： 制御パラメータ（個体数 K，節点率 λ）最終世代数，試行回数，スプラインの次数，最小突然変異率を入力する．

ステップ 3： 乱数を用いて個体の初期集団を生成する．

ステップ 4： 第 1 世代目の計算を行う．すなわち，各個体ごとに，その遺伝子列を内部節点としてスプラインによるあてはめを行い，評価値 e を計算する．また，最も良い個体を保存しておく．

ステップ 5： 3.3 節で述べた方法により個体の選択を行い，次世代の個体候補を生成する．

ステップ 6： 各個体ごとに，その遺伝子列を内部節点としてスプラインによるあてはめを行い，評価

値 e を計算する．また，最も良い評価値 $e_{\min}$ を求め，平均の評価値 $\bar{e}$ を計算する．

ステップ 7： 3.4 節で述べた方法により交叉を行い，次世代の個体候補を生成する．

ステップ 8： 各個体ごとに，その遺伝子列を内部節点としてスプラインによるあてはめを行い，評価値 e を計算する．また，最も良い評価値 $e_{\min}$ を求め，平均の評価値 $\bar{e}$ を計算する．さらに，最も良い個体を保存しておく．

ステップ 9： 3.5 節で述べた突然変異を行い，次世代の個体を生成する．

ステップ 10： 各個体ごとに，その遺伝子列を内部節点としてスプラインによるあてはめを行い，評価値 e を計算する．また，最も良い評価値 $e_{\min}$ を求め，平均の評価値 $\bar{e}$ を計算する．さらに，最も良い個体を保存しておく．

ステップ 11： ステップ 8 の最も良い個体とステップ 10 の最も良い個体を比較して，良い方をこの世代の最良個体とする．また，それと前世代までの最良個体を比較して良い方を保存しておく．

ステップ 12： 最終世代まで計算したか？ YES のとき，これまでに得られた最良個体とそれに対応するデータあてはめの結果を出力して計算を終了する．NO のときステップ 5 へ戻る．

上記のアルゴリズムの中では，ステップ 4，6，8，10 の「あてはめの計算」部分に計算負荷が集中しているが，必要であればこの部分は並列計算が可能である．また，本章で提案するアルゴリズムを実行するためには，ステップ 2 で述べたパラメータの設定が必要である．参考までに，その推奨値を述べておく．個体数 K：50〜100 の偶数，最終世代数：100〜300，試行回数 20〜40，階数 m：4〜6（3 次〜5 次のスプライン），最小突然変異率：0.01．なお，節点率 λ は，3.1 節で述べたようにして決める．

5. 節点多重化アルゴリズム

一般に，GA は大域探索には優れているが局所探索には適していないといわれている[24]．このため，多重節点を得る手法を 4 章の「GA を用いたデータあてはめのアルゴリズム」の中に埋め込まないで，別の最適化アルゴリズムとして提案し，それを “節点多重化アルゴリズム” と呼ぶことにする．

節点多重化アルゴリズムは，4 章のアルゴリズムで得られた準多重節点を多重節点化するものであり，その概要は以下のとおりである（**図 3** 参照）．今，ある準多重節点の中で，両端にある節点の間の距離を “節点間距離” と呼ぶことにして，その許容値を ω と書くことにする．

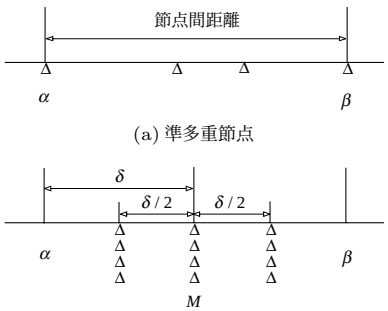


図 3 準多重節点の多重節点化（多重度 $r = 4$ の例）
Fig. 3 Multiplication of quasi-multiple knots.

ステップ 1： 内部節点の中で，節点間距離が許容値 ω 以下である節点群（準多重節点）を選び出す．

ステップ 2： その節点群に含まれる節点の数（r とする）を節点の多重度とする．また，その節点群の両端の値を α および β とする．さらに，$\delta = (\beta - \alpha)/2$ とおく．

ステップ 3： ステップ 1 で選び出された節点群を取り除き，代わりに区間 $[\alpha, \beta]$ の中点 M に r 重節点を置き，データあてはめを計算する．

ステップ 4： r 重節点を $M + \delta/2$ および $M - \delta/2$ へ移動させた場合について，データあてはめをそれぞれ再計算する．

ステップ 5： 3 つのデータあてはめ（M，$M + \delta/2$ および $M - \delta/2$ へ多重節点を置いた場合）の中で，最も良いものに対する r 重節点の位置を新しい M とする．

ステップ 6： $\delta \leq 0.01(\beta - \alpha)$ となれば，そのときの M を r 重節点の最適な位置として計算を終了する．そうでなければ，$\delta = \delta/2$ としてステップ 4 へ戻る．

ステップ 7： 節点多重化アルゴリズムを適用した結果，評価値が小さくなれば多重化後の結果を採用する．そうでなければ，多重化しない場合の結果を採用する．ただし，あらかじめ多重化すべきことが分かっていれば，それを優先する．

なお，節点間距離が許容値 ω 以下である節点群が複数箇所ある場合には，各箇所に対して上記のアルゴリズムを適用する．節点間距離の許容値 ω の値は $0.01(b - a)$ 程度にすればよい．この節点多重化アルゴリズムを適用した例は，6 章の例題 2 および例題 4 で示している．

6. 数値実験

本論文で提案するアルゴリズムの有効性を調べるため，多くの例題を用いて数値実験を行った．その中から，4つの例題の結果を報告する．例題1は，データの元にある関数に不連続なところや尖ったところがない “なめらかな” データであるが，例題2〜4はそのような “特異点” を含むデータである．

これらのデータは，上記の特性を持った関数に乱数で発生させた誤差を乗せて作成した．その誤差 ϵ_j は，すべて期待値0，分散1で正規分布をするものである．このため，最小二乗近似を計算するときの重み w_j の値はすべて1とした．

スプラインの次数は，最もよく使われている3次の場合について計算したが，本論文で提案する方法は次数には依存しないことに注意されたい．なお，以下の各図の「あてはめの結果」にある△印は節点の位置を表している．

例題1：データの元にある関数がなめらかな場合

あてはめるべきデータを次の式

$$F_j = 90/(1+e^{-100(x_j-0.4)}) + \epsilon_j, \quad (j=1,2,\cdots,N) \qquad (12)$$

で作成した．このデータは，$x=0.4$ 付近でステップ関数のように急に立ち上がっているが，データの元にある関数はなめらかであり，特異点は持っていない．ここで，横座標 x_j の値は $0.0, 0.005, \cdots, 1.0$ の201個，あてはめを行う区間は $[a,b]=[0,1]$ とした．制御パラメータは，個体数 $K=50$，節点率 $\lambda=0.025$ とした．したがって，内部節点（以下，簡単のため節点と呼ぶ）の数 n の初期値は5である．

図4 は，世代に対する評価値と内部節点の数（以下，簡単のため節点数と呼ぶ）の変化を示す計算結果である．細い実線，鎖線および点線は評価値を示している．細い実線は，最適（Best）な評価値を与えた試行結果であり，30世代目で収束している．鎖線は，初期集団を変えて30回の試行を行い，その平均（Average）をとったものである．また，点線は最悪（Worst）の評価値を与えた試行結果である．

さらに太い実線は，最適（Best）な評価値を与えた試行結果について，節点数（Knots）の変化を示したものである．節点数は，第1世代では5個であるが，世代が進むにつれて一度多くなった後減少していき，最終的には4個になっている．

節点数が，5個から4個へと単調に変化せず，一度多くなった後で減少する理由は以下のとおりである．

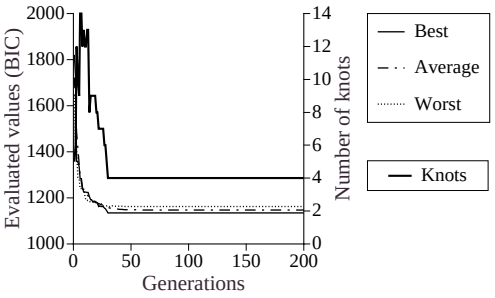


図4 例題1の評価値と節点数

Fig. 4 Evaluated values and number of knots for example 1.

節点の数が少なく，その位置が適切でないときには，一般に節点の多い個体の方が評価値が良い．そこで，モデル関数がデータの元にある関数をだいたい表現できるようになるまでは，最適な評価値の地位は節点の多い個体が占める．しかしその後は，節点位置を調整されたモデル関数の中に評価値がさらに良いものが現れ，節点数の少ない個体が最適な評価値の地位を奪うようになる．

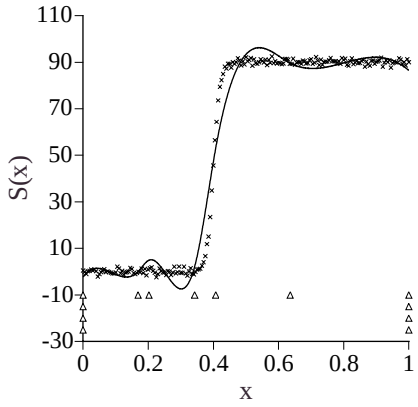
図5 は，データあてはめの結果である．図5 (a) は第1世代で最も良いものであるが，節点の数と位置が調整されていない．このため，曲線に大きなうねりが現れている．図5 (b) は収束したときであるが，関数 $S(x)$ はデータをよく近似しており，うねりを生じていない．このとき節点は，データの元にある関数の変化が大きいところに集中している．このことは，専門家の経験的な知識[2),5)]とよく一致している．**図6** は，30回の試行の中で最悪の場合である．この例題では，図5 (b) に示す最良の結果と大きな違いは見られない．

なお，本例題で200世代の計算を行ったときの計算時間は，シリコングラフィックス製のOrigin 2000（MIPS R10000 × 12CPU，195 MHz）を用いた場合約22.3秒であった．ただし，並列計算は行っていない．またこのとき，プログラムのプロファイル解析を行った結果，遺伝的操作に要する計算時間は，全体の計算時間に対して1%未満であった．

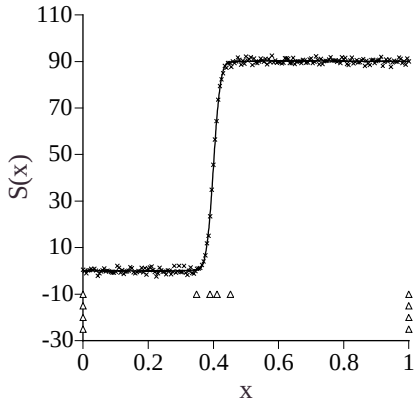
例題2：データの元にある関数が不連続点を持つ場合

あてはめるべきデータを次の式

$$F_j = \begin{cases} 1.0/\{0.01+(x_j-0.3)^2\}+\epsilon_j, & (x<0.6) \\ 1.0/\{0.015+(x_j-0.65)^2\}+\epsilon_j, & (0.6\le x) \end{cases} \quad (j=1,2,\cdots,N) \qquad (13)$$



(a) 第 1 世代目で最も良いもの



(b) 収束後

図 5 例題 1 のデータあてはめの結果
Fig. 5 Result of data fitting for example 1.

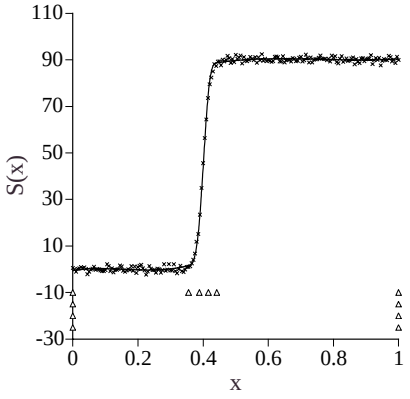


図 6 例題 1 のデータあてはめの結果（最悪の試行結果）
Fig. 6 Result of data fitting for example 1 (The worst case).

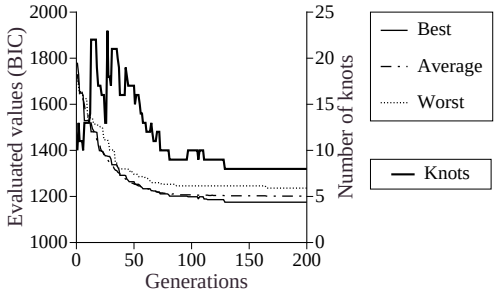


図 7 例題 2 の評価値と節点数
Fig. 7 Evaluated values and number of knots for example 2.

で作成した．このデータの元にある関数は，$x = 0.6$ で不連続な形をしている．ここで，横座標 x_j の値は $0.0, 0.005, \cdots, 1.0$ の 201 個とした．また，あてはめを行う区間は $[a, b] = [0, 1]$ とした．制御パラメータは，個体数 $K = 50$，節点率 $\lambda = 0.05$ とした．したがって，節点数 n の初期値は 10 である．

図 7 は，世代に対する評価値と節点数の変化を示す計算結果である．細い実線は，最適な評価値を与えた試行結果であり，129 世代目で収束している．収束までの世代数を例題 1 と比較すると，この例題の方が多い．その理由は，不連続点に節点が 4 個集ったモデルを探索することが容易でないからである．

また太い実線は，そのときの節点数の変化を示している．節点数は，第 1 世代では 10 個であるが，世代が進むにつれて振動しながら一度多くなった後減少していき，最終的には 8 個になっている．すなわち，ここでも例題 1 と同様な傾向が見られる．

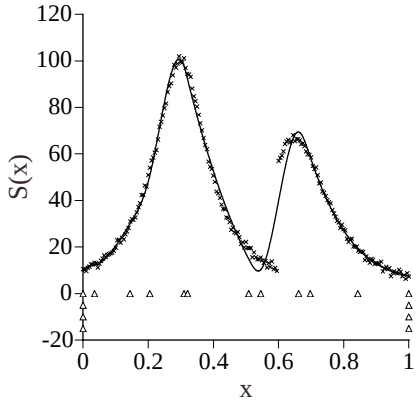
図 8 は，データあてはめの計算結果である．図 8 (a) は，第 1 世代で最も良いものであるが，節点の位置が調整されていないため，曲線に大きなうねりが現れている．また図 8 (b) は収束したときであるが，関数 $S(x)$ はデータをよく近似しており，うねりを生じていない．このとき節点は，不連続点 $x = 0.6$ に 4 個集中しており，準多重節点となっている．なお，3 次スプラインの場合，節点が 4 個集まればその点で関数値が不連続になることに注意されたい[12]．

図 9 は，5 章で述べた節点多重化アルゴリズムを用いて，図 8 (b) の $x = 0.6$ 付近の準多重節点を多重節点化したものである．このとき近似曲線は，$x = 0.6$ 付近で 4 重節点を持ち，その上で不連続となっている．また，評価値は 1175.3325 から 1175.2125 へ減少した．

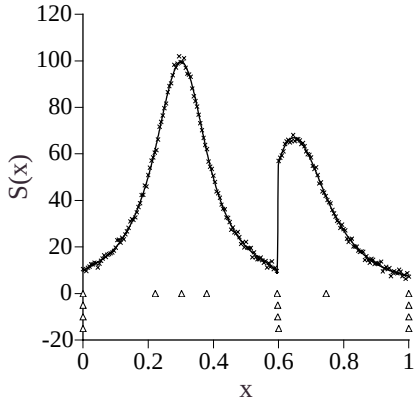
例題 3：データの元にある関数が尖った点を持つ場合

あてはめるべきデータを次の式

$$F_j = 100/e^{|x_j - 5|} + (x_j - 5)^5/500 + \epsilon_j, \quad (j = 1, 2, \cdots, N) \qquad (14)$$



(a) 第 1 世代目で最も良いもの



(b) 収束後

図 8 例題 2 のデータあてはめの結果
Fig. 8 Result of data fitting for example 2.

で作成した．このデータの元にある関数は，$x = 5$ で鋭く尖っている．ここで，横座標 x_j の値は $0.0, 0.05, \cdots, 10.0$ の 201 個，あてはめを行う区間は $[a, b] = [0, 10]$ とした．制御パラメータは，個体数 $K = 50$，節点率 $\lambda = 0.05$ とした．したがって，節点数 n の初期値は 10 である．

図 10 は，世代に対する評価値と節点数の変化を示す計算結果である．細い実線は，最適な評価値を与えた試行結果であり，30 世代目で収束している．また太い実線は，そのときの節点数の変化を示している．節点数は，第 1 世代では 10 個であるが，世代が進むにつれて振動しながら減少していき，最終的には 5 個になっている．

図 11 は，データあてはめの計算結果である．図 11 (a) は，第 1 世代で最も良いものであるが，$x = 5$ に節点が十分集中していないため，その点で曲線が丸くなっている．また図 11 (b) は収束したときであるが，関数 $S(x)$ はデータをよく近似しており，$x = 5$

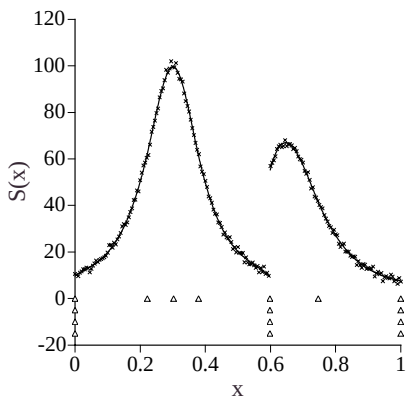


図 9 例題 2 のデータあてはめの結果（準多重節点の多重化後）
Fig. 9 Result of data fitting for example 2 (After multiplication of quasi-multiple knots).

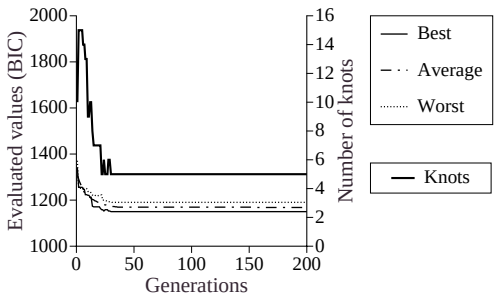


図 10 例題 3 の評価値と節点数
Fig. 10 Evaluated values and number of knots for example 3.

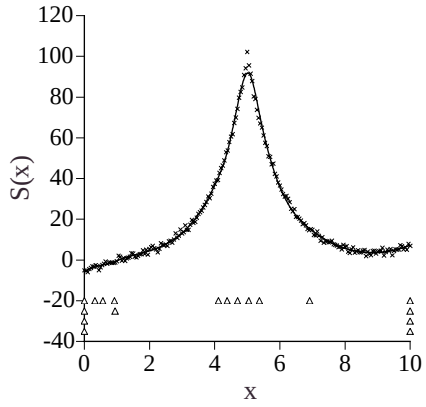
で鋭く尖っている．このとき節点は，$x = 5$ 付近に 3 個集中しており，準多重節点となっている．なお，3 次スプラインの場合，節点が 3 個集まればその点で微係数が不連続になることに注意されたい[12]．

例題 4：データの元にある関数が不連続点と尖った点の両方を持つ場合

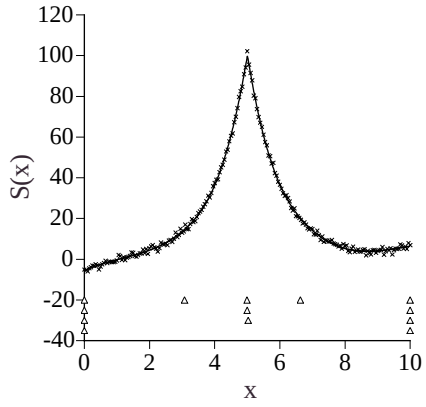
あてはめるべきデータを次の式

$$F_j = \begin{cases} 1.0/\{0.01 + (x_j - 0.9)^2\} + \epsilon_j, \\ \qquad\qquad (x < 1.0) \\ 100 \sin\{2\pi(x_j - 1.0)\} + \epsilon_j, \\ \qquad\qquad (1.0 \le x_j < 1.5) \\ 70|\sin\{2\pi(x_j - 1.0)\}| + \epsilon_j, \\ \qquad\qquad (1.5 \le x_j) \end{cases} \quad (j = 1, 2, \cdots, N) \quad (15)$$

で作成した．このデータの元にある関数は，$x = 1.0$ で不連続，$x = 1.5$ で尖っている．ここで，横座標 x_j の値は $0.0, 0.01, \cdots, 2.0$ の 201 個とした．また，あてはめを行う区間は $[a, b] = [0, 2]$ とした．制御パラ

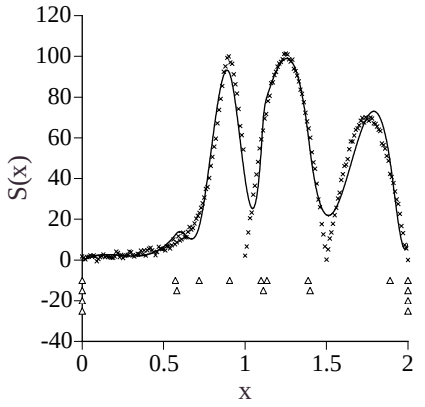


(a) 第 1 世代目で最も良いもの

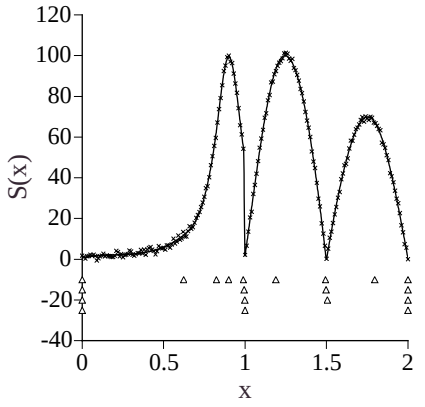


(b) 収束後

図 11 例題 3 のデータあてはめの結果
Fig. 11 Result of data fitting for example 3.



(a) 第 1 世代目で最も良いもの



(b) 収束後

図 13 例題 4 のデータあてはめの結果
Fig. 13 Result of data fitting for example 4.

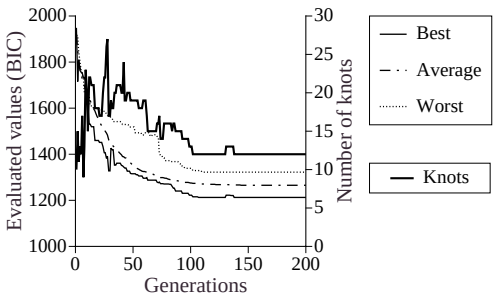


図 12 例題 4 の評価値と節点数
Fig. 12 Evaluated values and number of knots for example 4.

メータは，個体数 $K = 50$，節点率 $\lambda = 0.05$ とした．したがって，節点数 n の初期値は 10 である．

図 12 は，世代に対する評価値と節点数の変化を示す計算結果である．細い実線は，最適な評価値を与えた試行結果であり，138 世代目で収束している．収束までの世代数を例題 1，例題 3 と比較すると，この例題の方が多い．その理由は，不連続点と尖った点の両方を持つ関数形であるため複雑であるからである．

また太い実線は，そのときの節点数の変化を示している．節点数は，第 1 世代では 10 個であるが，世代が進むにつれて振動しながら一度多くなった後減少していき，最終的には 12 個になっている．10 個から 12 個へと単調に増加せず，一度多くなった後で減少する現象は，例題 1〜2 と同様である．

図 13 は，データあてはめの計算結果である．図 13 (a) は，第 1 世代で最も良いものである．この図を見ると，$x = 1.0$ と $x = 1.5$ 節点が十分集中していないため，不連続なところや尖ったところをうまく表現できていない．また，節点数が少なく位置も調整されていないため，曲線に大きなうねりが現れている．図 13 (b) は収束したときであるが，関数 $S(x)$ はデータをよく近似している．すなわち，$x = 1.0$ での不連続な形，$x = 1.5$ での尖った形をうまく表現できている．このとき節点は，$x = 1.0$ 付近に 4 個，$x = 1.5$

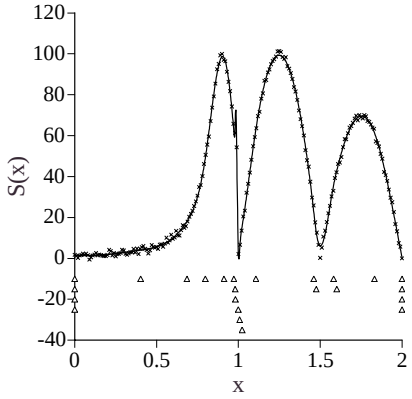


図 14 例題 4 のデータあてはめの結果（最悪の試行結果）
Fig. 14 Result of data fitting for example 4 (The worst case).

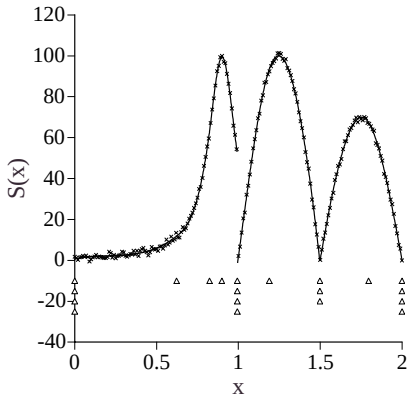


図 15 例題 4 のデータあてはめの結果（準多重節点の多重化後）
Fig. 15 Result of data fitting for example 4 (After multiplication of quasi-multiple knots).

付近に 3 個，それぞれ集中しており準多重節点となっている．

図 14 は，30 回の試行の中で最悪の場合である．この例題では，図 13 (b) に示す最良の結果と比べると，大きく異なっており，節点がうまく配置されていない．これは，初期収束を起こしているためであると思われる．

図 15 は，5 章で述べた節点多重化アルゴリズムを用いて，図 13 (b) の $x = 1.0$ および $x = 1.5$ 付近の準多重節点を多重節点化したものである．このとき近似曲線は，$x = 1.0$ 付近の 4 重節点上で不連続に，$x = 1.5$ 付近の 3 重節点上で関数値のみ連続になっている．また，評価値は 1212.6251 から 1212.6055 へ減少した．

7. あとがき

本論文では，スプラインを用いたデータあてはめの節点を，実数を遺伝子とした GA と情報量規準 BIC によって決定する方法を提案した．この方法の特徴は以下の 4 点である．

(1) 節点の位置が離散化の誤差の影響を受けない．したがって，文献 10)，11) の方法よりも節点の位置が正確に求まる．また，単一節点だけでなく準多重節点も扱うことが可能である．

(2) データの元にある関数が，なめらかなデータだけでなく，不連続なところや尖ったところのあるデータも扱うことが可能である．これは文献 10)，11) の方法では困難であったことであり，扱えるデータの範囲（すなわち，モデル関数のクラス）が広くなった．

(3) BIC の意味で最適なモデルを自動的に選択することによって，適切な節点の数と位置を自動的に決定できる．このため，従来の方法で必要とされることが多い許容誤差とか平滑化パラメータなどの設定は不要である．また，最適解を探索するために節点の良い初期値を与える必要がない．

(4) 交叉率および突然変異率を適応的に決定する方法を導入した．このため，これらの値をユーザが与える必要がないので使いやすい．

また，節点多重化アルゴリズムを提案することにより，準多重節点を多重節点化することを可能とした．

なお今後の課題としては，実際の計測データを用いて有効性を検証すること，平面データや多次元データへ適用できるように拡張すること，などがある．

謝辞 本論文に対して建設的なコメントをくださった査読者に感謝する．本研究の一部は文部省科学研究費補助金基盤研究 B（課題番号 10558052）の助成を受けた．

参考文献

1) 市田浩三，吉本富士市：スプライン関数とその応用，p.220, 教育出版 (1979).
2) Dierckx, P.: *Curve and Surface Fitting with Splines*, p.285, Clarendon Press-Oxford (1993).
3) Powell, M.J.D.: Curve fitting by splines in one variable, *Numerical Approximation to Functions and Data*, Hayes, J.G. (Ed.), Athlone Press, London (1970).
4) Ichida, K., Yoshimoto, F. and Kiyono, T.: Curve fitting by a piecewise cubic polynomial,

Computing, Vol.16, No.4, pp.329–338 (1976).
5) Cox, M.G.: A survey of numerical methods for data and function approximation, *The State of the Art in Numerical Analysis*, Jacobs, D.A.H. (Ed.), pp.627–668, Academic Press, New York (1977).
6) Jupp, D.L.B.: Approximation to data by splines with free knots, *SIAM J. Numer. Anal.*, Vol.15, No.2, pp.328–343 (1978).
7) Lyche, T. and Mørken, K.: A data-reduction strategy for splines with applications to the approximation of functions and data, *IMA Journal of Numerical Analysis*, Vol.8, No.2, pp.185–208 (1988).
8) Anthony, H.M., Cox, M.G. and Harris, P.M.: The use of local polynomial approximations in a knot-placement strategy for least-squares spline fitting, *NPL Report*, DITC 148/89 (1989).
9) 馬渡鎮夫，隆 雅久，豊田吉顯：スプライン平滑化における節点の自動設定に関する一考察，電子情報通信学会論文誌（D-II），Vol.J72-D-II, No.11, pp.1816–1823 (1989).
10) 吉本富士市，森山真光：スプライン関数を用いたデータあてはめ—遺伝的アルゴリズムによる節点の自動的な決定，情報処理学会論文誌，Vol.39, No.9, pp.2572–2580 (1998).
11) Yoshimoto, F., Moriyama, M. and Harada, T.: Automatic knot placement by a genetic algorithm for data fitting with a spline, *Shape Modeling International'99*, pp.162–169, IEEE Computer Society Press (1999).
12) 吉本富士市：スプライン—なめらかで柔軟な形状表現，bit 別冊「インターネット時代の数学」，戸川隼人ほか（編），pp.203–214, 共立出版 (1997).
13) Schwarz, G.: Estimating the dimension of a model, *The Annals of Statistics*, Vol.6, No.2, pp.461–464 (1978).
14) 松嶋敏泰：統計モデル選択の概要，オペレーションズ・リサーチ，Vol.41, No.7, pp.369–374 (1996).
15) Akaike, H.: A new look at the statistical model identification, *IEEE Trans. Automatic Control*, Vol.AC-19, No.6, pp.716–723 (1974).
16) de Boor, C.: *A Practical Guide to Splines*, p.392, Springer-Verlag (1978).
17) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, p.412, Addison-Wesley (1989).
18) 樋口哲也，北野宏明：遺伝的アルゴリズムとその応用，情報処理，Vol.34, No.7, pp.871–883 (1993).
19) 伊庭斉志：遺伝的アルゴリズムの基礎—GA の謎を解く，p.254, オーム社 (1994).
20) 坂和正敏，田中雅博：遺伝的アルゴリズム，p.203, 朝倉書店 (1995).
21) 山西健司，韓 太舜：MDL 入門：情報理論の立場から，人工知能学会誌，Vol.7, No.3, pp.427–434 (1992).
22) 小長谷明彦：確率的アプローチによる遺伝子情報処理，人工知能学会誌，Vol.8, No.3, pp.427–438 (1993).
23) Srinivas, M. and Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms, *IEEE Trans. Systems, Man and Cybernetics*, Vol.24, No.4, pp.656–667 (1994).
24) 城戸 隆：遺伝的アルゴリズムを用いたハイブリッド探索，遺伝的アルゴリズム，北野宏明（編），pp.61–88, 産業図書 (1993).

(平成 11 年 1 月 13 日受付)
(平成 11 年 11 月 4 日採録)

吉本富士市（正会員）

1966 年岡山大学工学部電気工学科卒業．明石工業高等専門学校，和歌山大学教育学部を経て，現職は和歌山大学システム工学部教授，システム情報学センター長．工学博士．形状モデリング，遺伝的アルゴリズム，数値計算，画像処理等の研究に従事．共著書「スプライン関数とその応用」（教育出版）等．IEEE，電子情報通信学会，日本応用数理学会，日本計算工学会等会員．

原田 利宣（正会員）

1987 年九州芸術工科大学工業設計学科卒業．同年，マツダ株式会社入社．1990 年千葉大学大学院工学研究科修了．同年，日産自動車（株）入社，自動車デザイン開発，研究業務に従事．1993〜96 年千葉大学大学院自然科学研究科に国内留学．1996 年和歌山大学助手．1997 年同助教授，現在に至る．日本デザイン学会 1996 年度研究奨励賞受賞．博士（工学）．日本デザイン学会，日本ファジィ学会，感性工学会，形の科学会等会員．

森山　真光（正会員）

1991 年広島大学総合科学部総合科学科卒業．1993 年同大学大学院工学研究科修士課程修了．1996 年大阪大学大学院基礎工学研究科物理系専攻（情報工学分野）博士課程単位取得認定退学．同年和歌山大学助手．1999 年和歌山大学システム工学部講師現在に至る．博士（工学）．形状モデリング，コンピュータビジョン，画像処理等の研究に従事．電子情報通信学会会員．

吉本　芳英

1972 年生．1995 年東京大学理学部物理学科卒業．1997 年同大学大学院理学系研究科物理学専攻修士課程修了．現在，同大学院理学系研究科物理学専攻博士課程在学中．専門は計算物性物理学だが数値解析，並列処理にも関心がある．日本物理学会会員．