

遺伝的プログラミングを用いた関数最適化手法による 遺伝子ネットワークの S-system モデルの推定

松村 幸輝[†] 種田 洋[†] 木村 周平[†]

本稿は遺伝的プログラミングを用いた新しい関数最適化手法を提案し、この手法を用いて遺伝子ネットワークの S-system モデルのパラメータ推定を行った結果について述べる。関数最適化問題を解くための進化計算法として、これまで様々な実数値 GA が提案されているが、性能が向上するのにもなっていない面が複雑化している面は否定できない。本研究では、シンプルなアルゴリズムに基づく新しい進化計算法として、目的関数のパラメータ集合の各実数値要素を式の木で表現し、この木構造に対して遺伝操作を施すという遺伝的プログラミング手法を用いた関数最適化方法を考案した。まず、この最適化手法を各種のベンチマーク関数に適用して有用性を検証した結果、特に多峰性関数に対して有効であることが分かった。そして、遺伝子ネットワークの S-system モデルのパラメータ推定を行った結果、良好な解を得ることができた。

Inference of S-system Models of Genetic Networks by Function Optimization Using Genetic Programming

KOKI MATSUMURA,[†] HIROSHI OIDA[†] and SHUHEI KIMURA[†]

In this paper, we propose a new technique for the function optimization using genetic programming approach, and then apply it for the inference of the S-system models of genetic network. To solve a function optimization problem, various techniques using real-coded GA have been proposed. However those algorithms are too complicated. Then, we design a technique using the genetic programming as a simple estimation method. In this technique each element of multi-dimensional real vector is expressed by a tree of expression, and the tree structure is evolved by the genetic operations. This optimization technique is verified using benchmark functions. As a result, especially it shows good performances on multi-peak functions. And also, as the real problem the S-system parameters of genetic networks are inferred by applying to the observed time-series data of gene expression patterns. Consequently, the effectiveness of the technique is verified to be able to obtain the appropriate solutions.

1. はじめに

連続な探索空間での関数最適化問題は、自然科学、社会科学を問わず、様々な分野で観測される重要な問題の 1 つである。具体的な実問題としては、幅広い需要と用途がある工学分野での各種設計問題、化学反応モデル、制御理論などの非線形性を扱う数理モデルの最適化の問題など、多くの問題がある。特にその中で、本稿でも取り扱うバイオインフォマティクス分野における遺伝子ネットワークの推定問題は、最近のトピックス的な問題として多くの研究が行われている^{1),2)}。

このような問題を解く有力な方法として、進化システムとして並列解法が可能な遺伝的アルゴリズム(以

下、GA)の検討が進められている。これは、最適化の対象となる関数に含まれる複数のパラメータを実数値要素としてとらえて最適化を図るもので、実数値 GA として様々なアルゴリズムが提案されている^{1),3)}。

実数値 GA の解法は、多くの場合、交叉処理において種々の工夫が施され、探索効率の向上が試みられている³⁾⁻⁵⁾。ただ、効果的解探索を重視するあまり、アルゴリズムが複雑化していることは否めない。

そこで本研究では、できるだけシンプルな計算原理に基づく解法として、遺伝的プログラミング⁶⁾(以下、GP)を適用した新しい関数最適化手法を提案する。

GP を適用した関数最適化手法としては、基本的に関数そのものを木構造で表現し、これにより適切な式構造を探索する方法が提案されている²⁾。筆者らも同様な考えに基づいてファジィ制御での最適メンバシップ関数の導出に応用し、所定の成果を得ている^{7),8)}。

[†] 鳥取大学工学部

Faculty of Engineering, Tottori University

一方、本稿で提案する GP を用いた関数最適化手法は、目的関数の要素である各実数値を、基準とする定数値と四則演算子だけの式の木で表現し、この木構造に対して遺伝操作を施すことによって新たな解候補である実数値を創成するという、きわめてシンプルなアルゴリズムに基づく新しい進化計算手法⁹⁾である。

そして、その実問題への応用として、バイオインフォマティクスの分野で解明が進められている遺伝子ネットワークの相互作用推定への適応を試みる。

以下、まず本提案手法のアルゴリズムを記述し、次にベンチマーク関数を用いた数値実験を通して本手法の性能評価を行う。そして、遺伝子ネットワークの S-system モデルにおけるパラメータ推定を行い、その有効性について検討する。

2. GP による関数最適化手法

本提案手法の原理とアルゴリズムについて記述する。

2.1 計算原理と個体表現

目的関数のパラメータ集合の要素 (x_1, x_2, \dots, x_n) の最適値の組合せを見出すために、進化システムの 1 つである実数値 GA が種々検討されている^{1),3)-5)}。この解探索方法は、基本的には、各要素が互いに 2 点間あるいはそれ以上の要素間で直接、正規乱数などに基づいて新たな実数値を生成することによって、もとの解候補（親個体）とは異なる別の解候補（子個体）を導出するという手続きをとるものである。

一方、GP を用いた本提案手法は、個体構造として実数値要素で構成した配列構造を持つ従来の GA 手法とは異なり、図 1 に示すように、パラメータ集合（個体）の各実数値要素が式の木からなる 2 分木構造を持つようにし、各要素（式の木）を遺伝操作に基づいて進化させることによって最適化を実現するものである⁹⁾。具体的な手続きは以下のとおりである。

- (1) 目的関数のパラメータの 1 組の組合せ（パラメータ集合）を 1 個体とし、その要素である各実数値 (x_1, x_2, \dots, x_n) を、所定の値を持った基準となる複数個の実数 (a, b, c, \dots) （終端子）と四則演算子 $(+, -, \times, /)$ である関数（非終端子）のノードからなる式の木で表現する。
- (2) 各個体は、目的関数の要素数（次元数）と同じ数だけの式の木を持ち、これらの式の木に基づいて個体の評価を行う。その評価値をその個体の適応度とする。
- (3) 各要素を表す木構造のノードに対して交叉などの遺伝操作を施すことによって新たな解候補を

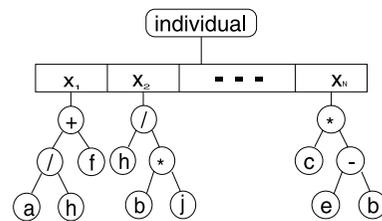


図 1 個体の構造

Fig. 1 Structure of individual.

創成するという手続きに基づいて最適化を行う。

ここで、終端子として実際に用いた数値は、種々値を変えて試行したが、最終的に 10 種類の実数値 $(0.5, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001, 0.00005, 0.00001)$ とした。なお、この定数集合には 0 を要素として含んでいない。このため、0 の値を表現するために若干木が大きくなること、あるいは 0 の持つ表現力を活かした解法を有効に利用できないこと、などの不利益があることは否めない。しかしながら、次章の数値実験の項でも述べるように、0 の値は比較的作成されやすいので、定数集合の要素として 0 が無いことで不都合が生じないこと、また 0 の値が他の実数値に比較して容易に出力されるという特殊性を排除することから、定数の要素群から 0 を省くものとした。

2.2 初期個体の生成

初期個体の生成は、個体の構成要素である式の木を、要素の数だけ作成することによって行う。詳細は以下のとおりである。

- (1) 要素の初期化領域内の値を乱数で発生させる。
- (2) 非終端子 1 つと終端子 2 つからなる高さ 2 の木を 50 個作成する。
- (3) 作成した木の値を計算し、その値が上述の乱数の値に近く、かつ初期化領域内であれば木の候補に加える。
- (4) 作成した木の総数が 1,000 を超えるか、木の候補が 20 を超えるまで次の操作を行う。
 - [1] 作成した木の中からランダムに 2 つを選び交叉を行う。
 - [2] 交叉により作成された木の値を計算し、発生させた乱数の値に近く、かつ初期化領域内であれば木の候補に加える。
 - [3] 上の [2] の交叉により作成された木の個数が 50 個になるごとに、はじめの乱数の値に近いものから順に、50 個を残し、(4) の操作を繰り返す。
- (5) 木の候補群からランダムに 1 つを選ぶ。ただし候補が 1 つもないときは高さ 2 の木の作成から

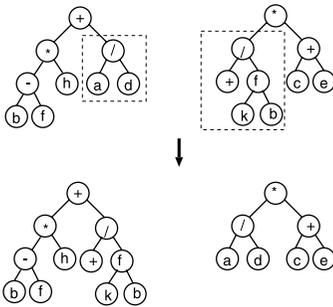


図 2 交叉
Fig. 2 Crossover.

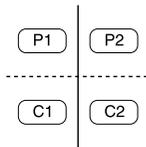


図 3 個体の入れ換え
Fig. 3 Exchange of individuals.

やり直す。

- (6) 上述の木の作成の工程を 1 個体の要素の数だけ繰り返し行って 1 つの初期個体を生成する。

以上の個体の生成の手続きを初期個体数分だけ繰り返して行うことによって、初期個体群を作成する。

2.3 交 叉

交叉は、図 2 に示すように、親個体に属する木から、その内容の一部を受け継ぐ子孫をつくり出す。

- (1) 個体群からランダムに 2 個体を選出する。
- (2) 選出された各個体に属する木のうちから交叉を行う木をランダムに選択する。
- (3) 選択された木の交叉を行う。その他の木はそのままコピーする。なお、交叉点は 1 点とする。
- (4) 交叉によってできた木の値を計算する。

なお、この交叉によってできる子個体は、交叉によって変異した 1 つの木だけが親個体と異なる構造となる。

2.4 進化過程での個体の入れ換えと世代

進化過程（解探索過程）において交叉を 1 回行うごとに優良個体の入れ換えを行う。この処理の様子を図 3 に示す。P1, P2 は親個体で、C1, C2 は交叉によってそれぞれ P1, P2 から変異した子個体である。

上述したように交叉によって生じる、P1 と C1（あるいは P2 と C2）の相異点は、交叉の行われた木が異なるだけで、残りの木は同じものである。このことから、個体の多様性が失われるのを防ぐために、P1 と C1 のいずれか、および P2 と C2 のいずれか、それぞれ適応度の高い方を次世代に残すようにした。

そして、このような 1 回の交叉とそれに続く個体の

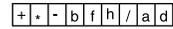


図 4 先順走査木の文字列
Fig. 4 Character string of pre-fixed tree.

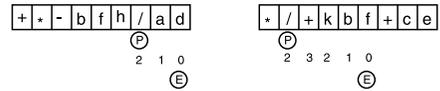


図 5 交叉処理の高速化
Fig. 5 Speed-up in crossover.

入れ換えで進化過程における 1 世代とした。

2.5 GP の高速化

GP の処理は、概念的には図 2 に示すようにリンク配置された木構造について行われるが、このような自己参照的構造体で構成される木構造に対してポインタ操作で遺伝操作手続きをすることはメモリ消費量が大きくなるとともに、処理速度の点においても非効率的な場合がある。このような観点から、木構造を一次元配列で表現した線形 GP 手法¹⁰⁾が提案されている。

以下は、本手法で考案した配列表現による高速化のためのデータ構造とアルゴリズムについて記述する。これにより、膨大な演算回数にも適用可能となる。

2.5.1 配列表現によるデータ構造

木構造のノード間の結合は通常、ポインタで表現するが、木の走査によってこれを配列を用いて実装する。木の走査には、ノードの走査順で 3 種類の走査方法があるが、ここでは処理の行いやすさを考慮して、親の走査後に左右の子を走査する先順走査方法を用いた。この例として、図 2 の左の親の木を先順走査したときに得られる文字列を配列で表現したものを図 4 に示す。

2.5.2 交叉処理のアルゴリズム

交叉処理は、図 5 のような手続きで行われる。配列で木構造を表現した場合、選ばれた交叉点から配列のどこまでが部分木であるかを解析する必要がある。そこで、次のようにして部分木を切り出す。

- (1) 交叉点 P を決定する。
- (2) 配列の P 番目の要素が非終端子であったならばそれに接続する子ノードの数を格納するスタック C（図中の数列）に子ノード数である 2 を代入し、終端子であったならば C に 0 を代入する。
- (3) 切り出す部分木の末尾を格納するスタック E に初期値として P を代入する。
- (4) C が 0 であれば E は切り出す部分木の最後の

要素となる。C が 0 でなければ C が 0 になるまで以下の手続きを繰り返し行う。

- [1] E に 1 を加え、配列要素を 1 つ進める。
- [2] E 番目の要素が非終端子ならば C に子の数 2 を加える。
- [3] C から取り出す子ノードの数 1 を引く。

なお、木のノード数の上限は 1,000 とし、交叉処理の段階で 1,000 を超えることが分かった時点で、その交叉処理は中断して、他のノードの選択からやり直す。

2.5.3 式の木の数値計算

個体の評価を行うために式の木の数値を算出する。図 6 に、先順走査による文字列から木の値を計算する過程を示す。なお、図中の T は終端子 (a = 1, b = 2, f = 6, d = 4), N は非終端子 (四則演算子) を表す。

2.5.4 性能比較

本節で述べた配列構造に基づく GP と通常の GP の比較実験を行った。通常の GP は、自己参照型構造体を用いてリンク配置によって 2 分木構造を実装し、ポインタ操作によって交叉処理を行うものである⁹⁾。

問題事例としては、次章で取り扱うベンチマーク関数の 1 つである Sphere 関数 (10 次元) (詳細は次章参照) を用い、これによる最適解探索問題に適応した。

実験条件と実験結果を表 1 に示す。同表に示されるように、実行時間、メモリ消費量 (通常 GP : 1 個体平均 198 ノード, 配列 GP : 要素数 1,000 (固定)) のいずれにおいても配列構造で個体表現した GP のほうが優れていることが分かる。

なお、使用機材は HP Compaq Business Desktop dx2000 ST (CPU : Pentium 4, 2.8 GHz, 1 MB

キャッシュ), OS は Turbolinux 10 とし, GCC で実装した。

3. 数値実験による性能評価

ベンチマーク関数を用いた数値実験を通して、本手法の性能を評価する。

3.1 目的関数

性能評価に用いるベンチマーク関数を表 2 に示す。また、2 次元の場合の概形と等高線図を図 7 に示す。

数値実験で用いる Sphere 関数は単純な単峰性関数であり、また offset Sphere 関数は、初期集団の初期化領域を Sphere 関数より狭い領域 ($5.11 \leq x_i \leq 5.12$) に限定したものである。この設定では集団が最適解を囲む領域には分布しておらず、稜構造を有する関数での GP の挙動を調べる実験となる。

また, Ackley 関数と Rastrigin 関数は多峰性の強い関数である。特に, Rastrigin 関数は大域的最適解のまわりに非常に多くの局所解を持つ関数である。この問題では初期集団の初期化領域を $-100 \leq x_i \leq 100 (i = 1, \dots, n)$ と通常よりも広くし, 難しい問題とした。

これらの関数において初期個体は, その構成要素である式の木の数値が表 2 に示す初期化領域内の値になるように限定して生成した。なお, 探索中は探索領域を明示的には考慮しないものとする。

数値実験を行うにあたって, それぞれの関数の次元数は 10 次元および 100 次元とした。また, 最適解の位置は, $(x_1, \dots, x_n) = (0, \dots, 0)$ だけでなく, 任意の値 (本実験では, $(x_1, \dots, x_n) = (0.368, \dots, 0.368)$)

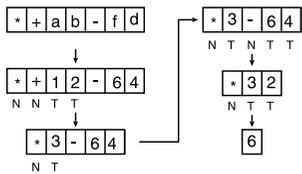


図 6 式の木の数値計算処理
Fig. 6 Calculation in tree of expression.

表 1 数値計算の実行比 (Sphere 関数の場合)

Table 1 Implementation ratio of math calculation in Sphere function.

	通常 GP	配列 GP
個体数	100	
世代数	1,000,000	
実行 CPU 時間	141.2 [s]	22.9 [s]
同比	6.2:1	
木 1 個あたりメモリ消費量	5.0 [KB]	1.0 [KB]
同比	5:1	

表 2 ベンチマーク関数

Table 2 Benchmark functions.

関数		初期化領域	最適解
Sphere 関数	$f_{sp}(x) = \sum_{i=1}^n x_i^2$	$-5.12 \leq x_i \leq 5.12$	$f_{sp}(0, \dots, 0) = 0$
offset Sphere 関数	$f_{osp}(x) = \sum_{i=1}^n x_i^2$	$5.11 \leq x_i \leq 5.12$	$f_{osp}(0, \dots, 0) = 0$
Ackley 関数	$f_{ac}(x) = 20 + e - 20 \exp \left(-0.2 \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}} \right) - \exp \left(\frac{\sum_{i=1}^n \cos 2\pi x_i}{n} \right)$	$-20 \leq x_i \leq 30$	$f_{ac}(0, \dots, 0) = 0$
Rastrigin 関数	$f_{ra}(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$	$-100 \leq x_i \leq 100$	$f_{ra}(0, \dots, 0) = 0$

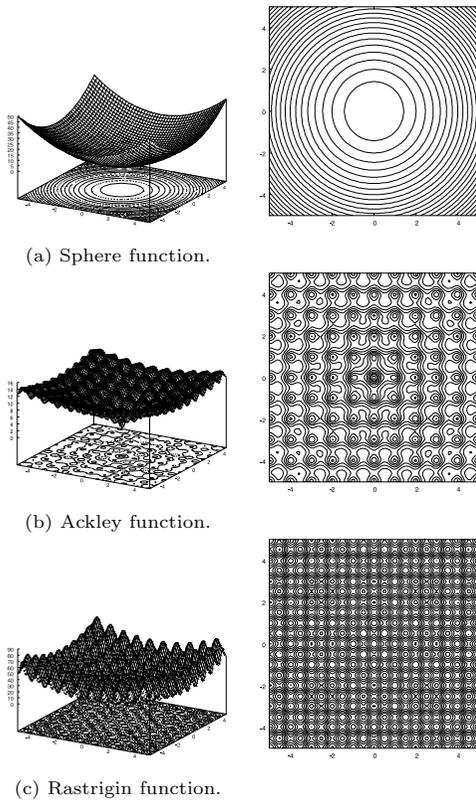


図 7 ベンチマーク関数の概形
Fig. 7 Shapes of benchmark functions.

となるように、 x_i の部分を $x_i - 0.368$ で置き換えたものについても実験を行った。

表 2 に示すように、最適解の値はいずれも 0 であるため、関数としての計算結果をそのまま評価値として用い、この値を個体の適応度とした。

GP 処理の条件としては、(1) 初期個体数：100 個体 (10 次元)、70 個体 (100 次元)、(2) 最大世代数 (総交叉回数)：10,000,000 (10 次元)、50,000,000 (100 次元)、(3) 個体の選択方式：ランダム選択、(4) 終了条件：適応度 (評価値) $\leq 10^{-6}$ 、とした。

3.2 結果と考察

数値実験の結果を示し、性能について考察する。

3.2.1 性能評価

世代推移による個体の適応度変化を図 8 に示す。計算条件としては、同図 (a) においては次元が 10 次元で、最適解の位置が 0、同様に図 (b) は 10 次元、0.368、図 (c) は 100 次元、0、図 (d) は 100 次元、0.368 とした。なお、計算評価値を適応度としているので、適応度の値は小さいほど良い評価となる。

これらの結果から、10 次元の場合では、いずれの関数も急速に最適解に収束していき、早い段階で最適

解の発見に至っていることが分かる。100 次元の場合でも同様に、世代が経るごとに最適化が進み、いずれも正しく最適解に収束している。なお、いずれの関数についても 10 次元と 100 次元の試行をそれぞれ 5 回ずつ行ったが、このすべての試行で最適解が発見できた。このように、10 次元から 100 次元に問題規模を拡張しても同様の計算過程で最適解を発見することが可能であることから、この手法は問題規模の変化に対するスケーラビリティに優れているものと考えられる。

また、単峰性関数だけでなく多峰性関数である Ackley 関数や Rastrigin 関数の場合にも最適解が得られたことは、良好な結果であったと考えられる。これは、本手法の遺伝操作手続きでできる木の値が元の値を中心として広い範囲に分布するような特徴を有するためと思われる。

ただ、最適解の位置を 0 とした場合 (同図 (a), (c)) と 0.368 とした場合 (同図 (b), (d)) で最適解の発見までの時間が異なる傾向がある。これは、終端子候補である定数の値が小さいので、乗算によって 0 近辺の値となりやすいことに基づく。最適解の位置によって性能変化が生じることは、この点に関してややロバスト性に欠けるものと思われるが、式の木の終端子の構成要素である定数の値を変えるなどの改善の余地があり、今後の検討課題となる。

3.2.2 従来手法との比較

他の実数値 GA 手法の結果と比較する。なお、以下の実験条件として、各関数の次元数は 100 次元とし、各パラメータの最適解の位置は 0 とした。

まず、図 9 は、Sphere 関数および offset Sphere 関数についての本手法による結果とあわせて、筆者らが行った種々の実数値 GA 手法による実験結果¹¹⁾を示している。Sphere 関数では、本手法は、UNDX-m+EDX/MGG 方式 (UNDX¹²⁾: Unimodal Normal Distribution Crossover, 多数の親を用いた単峰性正規分布交叉, EDX¹⁴⁾: Extrapolation-Directed Crossover, 外挿法型交叉, SPX/DDA 方式 (SPX⁴⁾: Simplex Crossover, シンプレックス交叉, DDA¹³⁾: Distance Dependent Alternation Model), SPX/MGG 方式 (MGG¹⁵⁾: Minimal Generation Gap, 最良 1 個体およびルーレット選択で選ばれた 1 個体を次世代に残す方式) などの高速な手法とほぼ同様な解探索過程を示している。また、offset Sphere 関数では、UNDX-m+EDX/MGG 方式に次いで良い解探索過程を示している。このように、これらの関数においては良好な結果が得られた。

次に、多峰性関数である Ackley 関数および Rast-

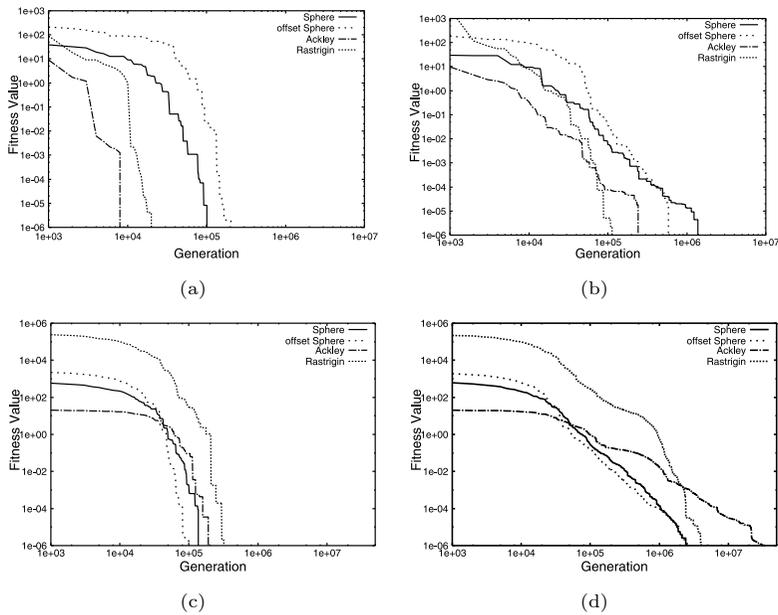


図 8 ベンチマーク関数による数値実験結果

Fig. 8 Experimental results of math calculations in benchmark functions.

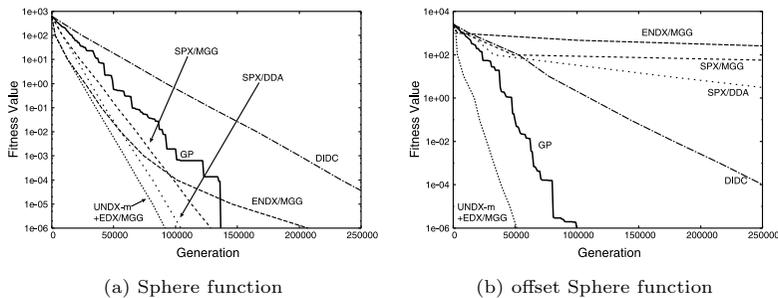


図 9 GA 手法との比較

Fig. 9 Comparison with GA techniques.

表 3 GA 手法との比較表

Table 3 Comparison table with GA techniques.

Objective function	Dimension n	GP this technique	GLSDC	DIDC	SPX/MGG	UNDX-m+EDX	IGA1	IGA8	IGA16
Ackley	10	8.00×10^3	3.60×10^2	2.60×10^4	3.26×10^4	1.21×10^4	1.17×10^4	1.16×10^4	1.18×10^4
	100	1.92×10^5	3.16×10^4	1.79×10^6	7.78×10^5	4.80×10^5	5.54×10^5	5.58×10^5	5.58×10^5
Rastrigin	10	2.00×10^4	1.08×10^3	2.58×10^4	3.30×10^4	2.06×10^4	1.78×10^4	1.74×10^4	1.83×10^4
	100	3.21×10^5	6.49×10^5	1.54×10^6	9.26×10^5	—	—	—	—

igin 関数についての本手法と他の GA 手法³⁾ による解探索結果をまとめたものを表 3 に示す. 表中の数値は, 適応度 (評価値) が 1.0×10^{-6} に達するまでの世代数 (交叉回数) である. 同表より, 最近の手法で高速である GLSDC 方式³⁾ (Genetic Local Search with distance independent Diversity Control, 並列化に適した遺伝的ローカルリサーチ) や DIDC 方式¹¹⁾ (Distance Independent Diversity Control, 距離に依存せずに多様性を制御し, 高次元の問題に適した GA)

と比較してもほぼ同様の良いパフォーマンスが得られた. なお, IGA1, IGA8, IGA16 はそれぞれ, 鳥数を 1, 8, 16 としたときの並列処理に適した島モデル GA³⁾ で, これらに対しても優れていることが分かる. このことから, 本手法は, 多峰性関数の最適化に有効な手段であると考えられる.

以上, 本手法をベンチマーク関数の最適化に適用した結果, 概して良好な結果が得られた. このことから, 実問題に適用可能な性能を有するものと思われる.

4. 遺伝子ネットワークの相互作用推定

本提案手法を、遺伝子ネットワークの S-system モデルに適応し、遺伝子間の相互作用を推定する。

4.1 S-system モデル

S-system は以下の連立微分方程式によって記述されるモデルである¹⁶⁾。

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^N X_j^{g_{i,j}} - \beta_i \prod_{j=1}^N X_j^{h_{i,j}}, \quad (i=1, \dots, N) \quad (1)$$

ここで、 X_i は要素 (遺伝子) i の状態変数、 N はネットワークに含まれる全要素数、 $g_{i,j}$ は X_i の生成過程に参与する X_j の相互作用係数、 $h_{i,j}$ は X_i の分解過程に参与する X_j の相互作用係数、また、 α_i 、 β_i はそれぞれ X_i の生成項、分解項に乘じる係数である。

S-system モデルは、ある状態変数 X_i の生成過程と分解過程に、システムを構成しているすべての状態変数 $X_j (j=1, 2, \dots, n)$ が関与していると仮定する全結線モデルである。このモデルは、複数の生成・分解過程からなる反応経路を近似した表現となっており、詳細な機構の明らかになっていない遺伝子ネットワークの記述に有効であると考えられている。

4.2 問題の定式化

一般に遺伝子ネットワークの S-system モデルを推定する問題は、以下の関数を最小化するような S-system パラメータ α_i 、 β_i 、 $g_{i,j}$ 、 $h_{i,j}$ を見出す関数最小化問題として定式化される。

$$f = \sum_{i=1}^N \sum_{t=1}^T \left(\frac{X_{\text{cal},i,t} - X_{\text{exp},i,t}}{X_{\text{exp},i,t}} \right)^2 \quad (2)$$

ここで、 $X_{\text{exp},i,t}$ は実験的に得られる時刻 t における遺伝子 i の遺伝子発現量、同様に $X_{\text{cal},i,t}$ は連立微分方程式 (1) を解くことによって得られる遺伝子発現量、 N はネットワーク中の要素数、 T は観測した時系列データのタイムポイント数である。

遺伝子ネットワーク推定問題において解くべき問題の次元数は $2N(N+1)$ である。そのため既存の非線形関数最適化器を用いて大規模な遺伝子ネットワークを推定することは難しい。このことから、S-system モデルに基づく遺伝子ネットワーク推定問題の高次元性を解決するために、問題をいくつかの部分問題に分割する方法が試みられている^{17),18)}。

この方法は、遺伝ネットワーク推定問題をいくつかの下位の問題 (各遺伝子に相当する) に分割することで実現できる。 i 番目の遺伝子に関する下位問題は、

次式で示される。

$$f_i = \sum_{t=1}^T \left(\frac{X_{\text{cal},i,t} - X_{\text{exp},i,t}}{X_{\text{exp},i,t}} \right)^2 \quad (3)$$

ここで、 $X_{\text{cal},i,t}$ は i 番目の遺伝子の時刻 t における遺伝子発現量としての計算値である。これは次式の微分方程式 (4) を解くことによって得られる。

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^N Y_j^{g_{i,j}} - \beta_i \prod_{j=1}^N Y_j^{h_{i,j}} \quad (4)$$

ただし、

$$Y_j = \begin{cases} X_j, & \text{if } j = i, \\ \hat{X}_j, & \text{otherwise} \end{cases} \quad (5)$$

\hat{X}_j は微分方程式の解ではなく観測された時系列データで直接評価により計算した遺伝子発現レベルである。

なお、この手続きは評価に 2 つの方法を用いている。データが測定誤差を含まない状態で観察されると仮定される場合では、 \hat{X}_j は観測された遺伝子発現レベルのスプライン補間¹⁹⁾を用いて評価し、また測定誤差の有無が確認できない場合は、局所線形回帰²⁰⁾を用いて評価する。

これにより、 $2(N+1)$ 個の S-system のパラメータが与えられる場合、式 (4) は解くことが可能となる。したがって、この分割によって、 $2N(N+1)$ の次元のネットワーク推定問題は N 個の $2(N+1)$ 次元の問題に変換できる。なお、正確な曲線が観測された遺伝子発現パターンとして与えられる場合、式 (4) の解は、同時に式 (1) の解として与えられる。

4.3 数値実験

S-system モデルの推定を行った結果を述べる。

4.3.1 目標ネットワークモデル

提案手法の有効性を実証するために、人工的に作成した遺伝子ネットワークモデルを使用した。

この目標ネットワークモデルとして、図 10 に 5 つの遺伝子からなる小規模な遺伝子ネットワークモデル²¹⁾を示す。同図 (a) は、2 つの遺伝子 (遺伝子 1 と遺伝子 4) から構成される典型的な遺伝子相互作用系を示したもので、同図 (b) はこれを簡易的に記述したものである。なお、同図 (a) 中の + は正の相互作用を、- は負の相互作用を表す。pool は各状態変数の合成過程のプール (一定値) を表す。また、 X_1 は遺伝子 1 から生成される mRNA、 X_2 はそれから生成される酵素タンパク質、 X_3 は X_2 が触媒する誘導タンパク質、 X_4 は遺伝子 4 から生成される mRNA、 X_5 はそれから生成される制御タンパク質であると仮定し

表 4 目標モデルの S-system パラメータ

Table 4 S-system parameters of the target model.

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	5.0	0.0	0.0	1.0	0.0	-1.0	10.0	2.0	0.0	0.0	0.0	0.0
2	10.0	2.0	0.0	0.0	0.0	0.0	10.0	0.0	2.0	0.0	0.0	0.0
3	10.0	0.0	-1.0	0.0	0.0	0.0	10.0	0.0	-1.0	2.0	0.0	0.0
4	8.0	0.0	0.0	2.0	0.0	-1.0	10.0	0.0	0.0	0.0	2.0	0.0
5	10.0	0.0	0.0	0.0	2.0	0.0	10.0	0.0	0.0	0.0	0.0	2.0

表 5 時系列データの例

Table 5 Example of time-series data.

time	X_1	X_2	X_3	X_4	X_5
0.00000e+00	1.655967e+00	1.868416e+00	1.032173e-01	2.730268e-01	1.562687e+00
2.00000e-02	1.253467e+00	1.657448e+00	2.135465e-01	2.619844e-01	1.201692e+00
4.00000e-02	1.022979e+00	1.435307e+00	3.333388e-01	2.599481e-01	9.795225e-01
6.00000e-02	8.871738e-01	1.255195e+00	4.587796e-01	2.742908e-01	8.312707e-01
8.00000e-02	8.116321e-01	1.118002e+00	5.816339e-01	3.137539e-01	7.763355e-01
1.00000e-01	7.789859e-01	1.016770e+00	6.927056e-01	3.845379e-01	6.567651e-01
1.20000e-01	7.758686e-01	9.454224e-01	7.850088e-01	4.853233e-01	6.142717e-01
1.40000e-01	7.895703e-01	8.983366e-01	8.557147e-01	6.040636e-01	6.005016e-01
1.60000e-01	8.076667e-01	8.700321e-01	9.060943e-01	7.210789e-01	6.153848e-01
1.80000e-01	8.207603e-01	8.545919e-01	9.400428e-01	8.179174e-01	6.542878e-01
2.00000e-01	8.250062e-01	8.460691e-01	9.620969e-01	8.858285e-01	7.075012e-01

表 6 目標モデルの初期遺伝子発現レベル

Table 6 Sets of the initial gene expression levels of the target model.

Set	X_1	X_2	X_3	X_4	X_5
1	1.655967e+00	1.868416e+00	1.032173e-01	2.730268e-01	1.562687e+00
2	7.862766e-01	5.474855e-01	9.287958e-01	3.894443e-01	9.344040e-01
3	3.468547e-01	1.994981e+00	1.532913e+00	1.761393e+00	1.264981e+00
4	8.020131e-01	8.949262e-01	3.135082e-01	7.610533e-02	1.269705e+00
5	9.590725e-01	2.805737e-01	5.507401e-01	1.684232e+00	5.744767e-01
6	3.992936e-01	1.849408e+00	2.912736e-01	1.144217e+00	9.988814e-01
7	1.055713e-02	5.114093e-02	8.495855e-01	1.740444e+00	1.969969e-01
8	1.489803e+00	9.168820e-01	1.707836e+00	1.827741e+00	2.824051e-01
9	1.842769e-01	1.589055e+00	6.668454e-01	4.727903e-01	1.265678e+00
10	1.285656e+00	8.995862e-01	1.994967e-01	8.811659e-01	1.723054e+00
11	1.336883e-01	4.233753e-01	4.168260e-01	4.823942e-01	5.539923e-01
12	1.652500e+00	1.744966e+00	3.904404e-01	1.584671e+00	4.339247e-01
13	1.562800e+00	1.164151e+00	1.391469e+00	6.808265e-01	1.090292e+00
14	3.271505e-01	1.147837e+00	1.576167e-01	8.64541e-01	2.591408e-01
15	5.522177e-01	4.220327e-01	1.084436e+00	1.994388e+00	1.050098e+00

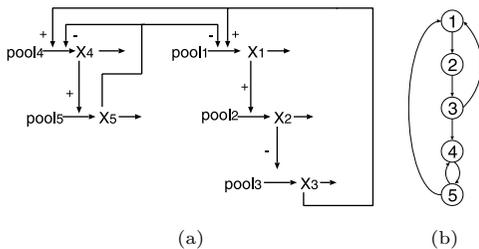


図 10 目標モデルのネットワーク構造

Fig. 10 Network structure of the target model.

ている。

この目標モデルのパラメータ（目標値）を表 4 に示す。このネットワーク推定問題は 4.2 節で述べた問題分割手法の適用によって 5 つの解問題に分割している。

このようなパラメータを関数最適化によって導出するためには、遺伝子発現パターンとして時系列データが必要となる。与えられる時系列データの量が十分でないと、S-system モデルの自由度は高くなり、解候補は増大する。このことから、適切な解を見出すために十分な量の遺伝子発現データとして、5 つの遺伝子すべてに対して 15 セットの時系列データを用意した。実際には、時系列データのセットは、生物学実験によって種々の異なる実験条件下で得るものであるが、本数値実験では、任意に生成した初期値と目標モデル上の微分方程式 (1) を用いて計算することによって得た。

時系列データは各遺伝子ごとに 11 サンプルポイント割り当て、これを 15 セット用意するので、各遺伝子上の観測時系列データは 165 サンプルポイントで構成する。例として第 1 セットの時系列データを表 5 に示す。なお、実際の計算ではこれらの値からスプライン補間法で求めた曲線を使用した。また、本実験で使用した全時系列データの初期値を表 6 に示す。

実験条件および各係数は、前報¹⁸⁾と同様、以下のように設定した。S-system の探索領域は、 α_i と β_i が 0.0 から 20.0、 $g_{i,j}$ と $h_{i,j}$ が -3.0 から 3.0 とした。また、計算コストを軽減するため、閾値 δ_s を設定し、 $g_{i,j}$ 、 $h_{i,j}$ の値がこの閾値よりも小さいとき 0 の値を返した。本実験では $\delta_s = 1.0 \times 10^{-3}$ を使用した。

なお、個体数は、標準で 1 分割問題のパラメータ数 12 個の 3 倍の 36 個とした。個体数による性能への影響に関しては、次節で詳述する。

4.3.2 結果と考察

本手法で得られた実験結果を示し、考察する。

4.3.2.1 最適化過程と S-system パラメータ

数値実験によって得られた適応度の推移を図 11 に示す。なお、同図 (a) ~ (e) は複数回実験を行ったときの各遺伝子ごとの適応度の推移、また同図 (f) は各遺伝子での適応度推移の最良のものを抽出して示している。これより、世代が進むにつれて、いずれの遺伝子

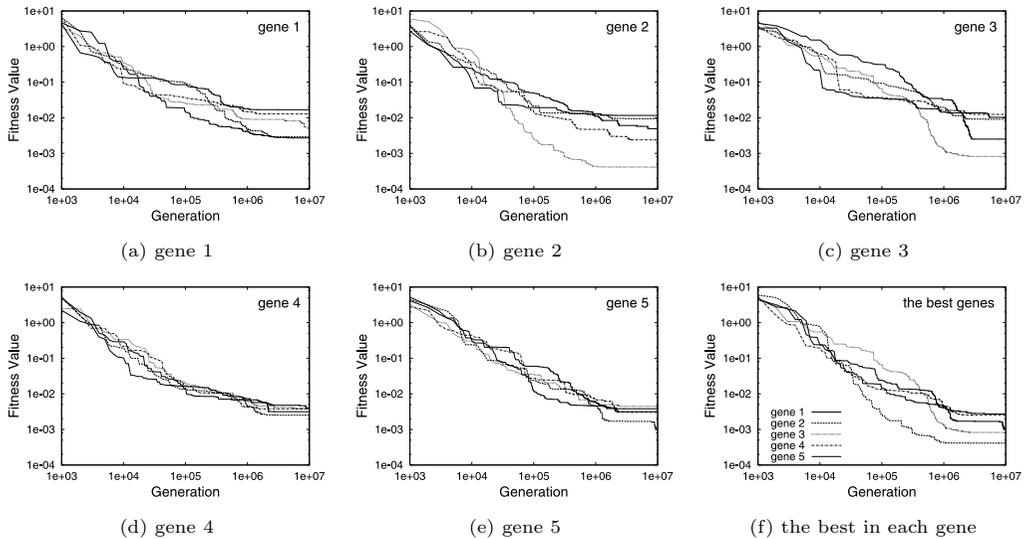


図 11 適応度の推移

Fig. 11 Transitions of fitness value.

表 7 数値実験により得られた S-system パラメータ

Table 7 Estimated S-system parameters for the target model.

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	4.600	-0.059	-0.010	0.967	0.001	-0.944	9.585	2.049	0.000	-0.040	0.015	0.100
2	9.813	2.029	-0.020	-0.011	0.006	0.000	9.816	-0.007	2.016	-0.005	0.001	0.000
3	9.137	-0.007	-0.969	-0.040	-0.009	-0.009	9.091	-0.006	-0.975	2.180	-0.025	-0.007
4	7.183	-0.066	0.064	2.080	-0.060	-1.065	9.145	-0.050	0.043	-0.067	2.100	0.030
5	9.256	0.006	0.010	0.036	2.096	-0.100	9.302	0.009	0.014	0.018	-0.050	2.065

に対する適応度も一様に改善されることが示され、これとともに S-system モデルパラメータの解探索が行われていることが示唆される。

この数値実験から得られた S-system パラメータの値を表 7 に示す。同表に示される各パラメータ値は、各遺伝子の目標値に近い値となっており、各遺伝子間の相互作用を推定するという所定の目的はほぼ達成されているものと考えられる。

一方、本手法とは異なる方式による GP 手法を用いた関数最適化法を提案した最近の報告^{2),22)}がある。その中で、同様の 5 遺伝子ネットワークの推定実験が行われているが、その S-system パラメータ値はきわめて良好な数値を示しており、ほぼ目標値と等しい値となっている。これと比較するため、目標値との近さを表す定量的な目安として、60 個のパラメータのそれぞれの相対誤差の絶対値の総和 σ を求めた。この値を用いて表すと、上記の他手法²⁾で 0.070、本手法で 1.185 となり、差異があることは否めない。

ただ、今回、本手法で得られたパラメータ値は、筆者らが先行して行った高性能な実数値 GA (GLSDC) による実験結果¹⁸⁾と比較しても、ほぼ同程度の値 (GLSDC 手法による σ の値は 0.366) となっている。

このことから、完全に正確なパラメータ評価は困難であるが、生物学的ネットワークを解釈するのに十分正確であるものと考えられる。

ここで、本手続きによって得られた各パラメータを構成する式の木の構造の一例を表 8 に示す。同表中の各要素は、表 7 に示す遺伝子 3 の各パラメータを表現する式の木について先順走査で得られた文字列で示している。これに示されるように、構成要素数は、少ないもので数個から多いもので 1,000 個程度となっている。また、具体的な式の木の構造の例として、遺伝子 3 の $g_{3,3}$ と $h_{3,3}$ を木構造で表したものを図 12 に示す。このように、高さ 2 の初期木から始まった式の木は、適応度評価を基にした遺伝操作によって進化し、進化過程でのランダム性や進化速度あるいは収束値などに依りて最終的には階層数や構成要素数などの構造表現の異なる式の木へと成長するのである。

これに関連して、上述の GP を用いた関数同定で、結果の良否と遺伝子長になんらかの相関関係があることが示されている²⁾。このことを考慮すれば、今回の実験では、構成要素数は上限を 1,000 とする緩やかな条件を付与したが、階層数については制限を設けなかった。上述のことを考慮すれば、より効率的な探索

表 8 木構造を表す先順走査文字列

Table 8 Prefixed character strings showing tree structure.

Table with 2 columns: 係数 (Coefficient) and 先順走査文字列 (Prefix character string). It lists various coefficients and their corresponding tree structures represented by strings of characters.

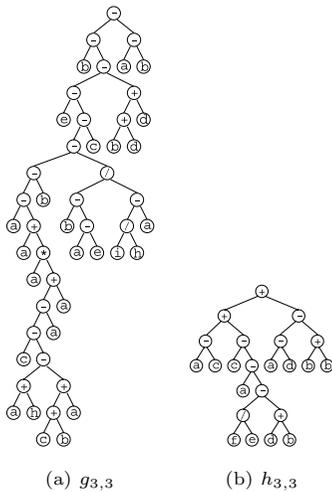


図 12 式の木の例

Fig. 12 Examples of expression tree.

過程を実現するためには、制限条件について検討の余地があるものと思われる。

4.3.2.2 ノイズデータによるネットワークの推定

実際に測定される時系列データには多くのノイズが含まれる。このノイズの影響を調べるため、ノイズを

含んだ時系列データを用いて数値実験を行った。

数値実験に用いた時系列データの例を図 13 (a) に示す。このサンプルデータは、第 1 データセットの遺伝子 1 の時系列データで、各サンプリングポイントでの時系列データの値にそれぞれ、1%、3%、5%、10%、20%のランダムノイズを付加したものである。

これに対する数値実験の結果を図 13 (b) ~ (e) に示す。同図はそれぞれ、上述のノイズ比に設定した場合に得られた世代に対する適応度の推移を示す。これより、ノイズが大きくなるにしたがって、ノイズがない場合（たとえば、図 11 (f)）に比較して、最終適応度は世代を経ても大きくは改善されにくくなる傾向があることが分かる。また、図 13 (f) はこれらの結果をまとめてノイズ比による最終適応度の変化を示したものである。これより、最終適応度はノイズに強く影響を受けることが分かる。

表 9 は、それぞれの場合で得られた S-system パラメータの値を示す。同表において、ノイズ比が大きいほど、目標値から外れる傾向が観察されるが、たとえ 20%程度の大きなノイズがあっても、各遺伝子の相互作用の存在を表すには可能な数値が得られており、この点においてロバスト性は高いものと思われる。この

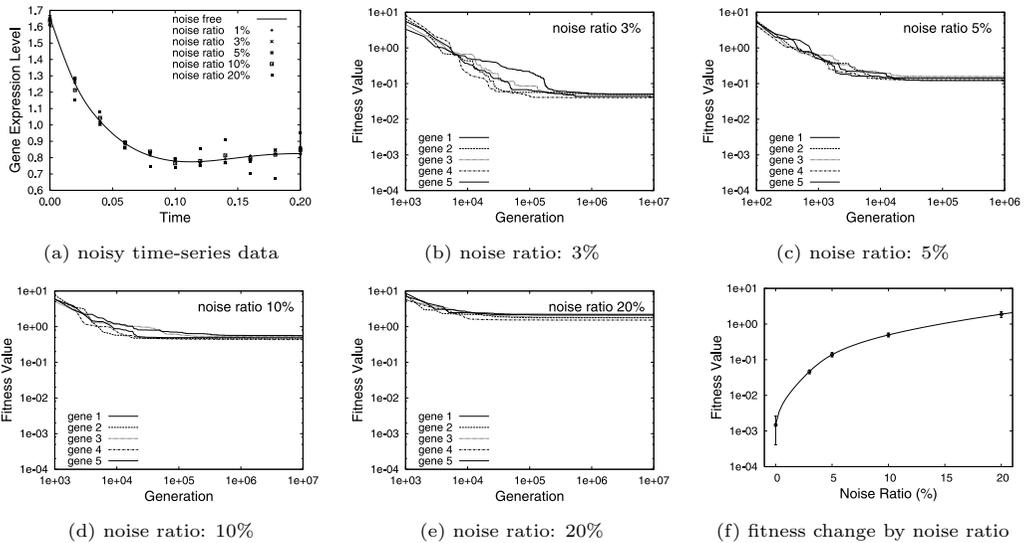


図 13 ノイズデータと適応度変化への影響

Fig. 13 Noisy data and influence to variation of fitness.

表 9 ノイズデータを用いた数値実験により得られた S-system パラメータ

Table 9 Estimated S-system parameters with noisy data.

(a) noise-free

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	4.600	-0.059	-0.010	0.967	0.001	-0.944	9.585	2.049	0.000	-0.040	0.015	0.100
2	9.813	2.029	-0.020	-0.011	0.006	0.000	9.816	-0.007	2.016	-0.005	0.001	0.000
3	9.137	-0.007	-0.969	-0.040	-0.009	-0.009	9.091	-0.006	-0.975	2.180	-0.025	-0.007
4	7.183	-0.065	0.064	2.080	-0.060	-1.065	9.145	-0.050	0.043	-0.067	2.100	0.030
5	9.256	0.006	0.010	0.036	2.096	-0.100	9.302	0.009	0.014	0.018	-0.050	2.065

(b) noise ratio: 3%

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	4.670	-0.080	0.050	1.225	-0.009	-1.081	9.852	2.094	0.057	0.076	-0.039	0.050
2	10.060	2.101	-0.040	0.099	0.040	0.010	10.000	-0.012	2.050	0.099	0.025	-0.029
3	18.206	0.000	-0.860	0.350	-0.005	0.001	18.357	0.010	-0.855	1.502	-0.011	0.005
4	7.349	0.037	-0.010	1.956	-0.010	-1.029	9.110	0.020	-0.022	-0.168	2.127	0.059
5	10.580	0.050	-0.023	0.037	1.975	-0.014	10.549	0.030	-0.009	0.048	0.015	1.950

(c) noise ratio: 5%

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	5.200	-0.029	0.010	0.995	0.037	-0.928	10.333	1.905	0.040	0.048	-0.001	0.080
2	7.669	2.225	-0.207	0.102	0.024	-0.014	7.500	-0.245	2.233	0.049	0.010	-0.060
3	6.841	-0.099	-0.949	-0.254	0.016	0.090	6.948	-0.058	-0.970	2.657	0.099	0.043
4	5.225	-0.098	0.100	2.290	-0.222	-1.300	7.202	-0.001	0.045	-0.395	2.457	0.205
5	10.600	0.000	0.000	0.050	1.963	0.010	10.500	0.010	-0.017	0.050	0.005	2.000

(d) noise ratio: 10%

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	3.765	-0.190	0.037	1.195	-0.004	-1.105	9.100	2.500	-0.100	-0.049	0.002	0.100
2	9.600	2.297	-0.131	0.009	-0.381	0.050	10.000	0.100	2.102	0.101	-0.305	0.000
3	6.400	0.000	-0.800	-0.333	0.011	-0.100	6.502	-0.004	-0.759	2.601	0.127	-0.138
4	3.601	0.050	-0.100	2.900	-0.508	-1.794	5.511	-0.101	-0.000	-0.504	2.950	0.100
5	10.005	0.001	0.050	0.059	2.100	0.001	10.511	0.101	0.025	0.080	-0.01758	2.183

(e) noise ratio: 20%

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	5.106	-0.145	0.0100	0.933	-0.004	-0.489	11.662	2.000	0.005	0.050	0.046	0.364
2	11.563	1.912	0.100	-0.101	0.450	-0.305	12.001	0.001	2.000	0.100	0.200	-0.095
3	6.678	-0.225	-0.711	-0.258	0.132	0.100	7.200	0.010	-0.850	2.001	0.441	-0.177
4	3.080	0.001	-0.400	2.950	-0.751	-1.500	5.506	-0.034	-0.245	-0.535	2.750	0.510
5	5.200	0.000	0.060	0.010	2.400	-0.550	5.511	-0.220	0.188	0.173	-0.562	2.958

ことから、ノイズが存在する状態であっても、本手法によって遺伝子間の相互作用を推定することが可能であると考えられる。

4.3.2.3 個体数変化による性能評価

個体数を種々変えた場合の性能評価を行った。図 14 に、個体数を 36 ~ 360 まで変えた場合に得られた実験結果の一部を示す。同図 (a) ~ (d) は、例として、個体数をそれぞれ、36, 72, 180, 360 とした場合の適応

度の推移を示す。同図 (e) は、個体数による適応度推移の変化を遺伝子ごとにまとめたもののうち、代表例として遺伝子 2 の場合の結果を示す。また、同図 (f) は、全実験結果に基づいてリプロットした、個体数による最終適応度の変化を示す、図中の直線は最小二乗法で導出したものである。

同図 (a) ~ (d) より、個体数が多い方が、推移曲線はより世代の大きい方にシフトしており、最終適応度に

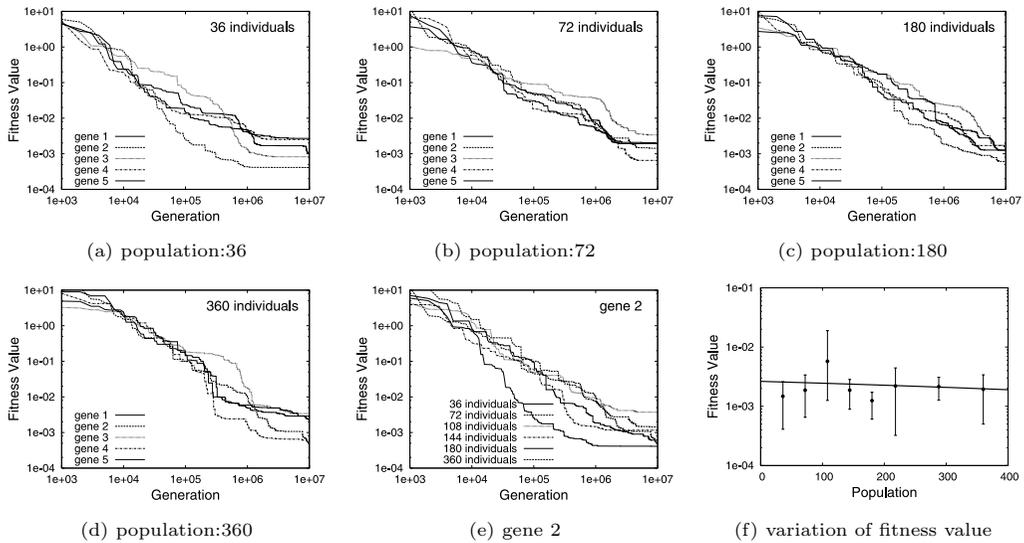


図 14 個体数による適応度の推移の変化

Fig. 14 Variations of transition of fitness by individual population.

表 10 個体数による S-system パラメータの変化

Table 10 Variation of estimated S-system parameters by individual population.

(a) individual population: 36

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	4.600	-0.059	-0.010	0.967	0.001	-0.944	9.585	2.049	0.000	-0.040	0.015	0.100
2	9.813	2.029	-0.020	-0.011	0.006	0.000	9.816	-0.007	2.016	-0.005	0.001	0.000
3	9.137	-0.007	-0.969	-0.040	-0.009	-0.009	9.091	-0.006	-0.975	2.180	-0.025	-0.007
4	7.183	-0.066	0.064	2.080	-0.060	-1.065	9.145	-0.050	0.043	-0.067	2.100	0.030
5	9.256	0.006	0.010	0.036	2.096	-0.100	9.302	0.009	0.014	0.018	-0.050	2.065

(b) individual population: 72

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	4.933	-0.043	0.027	1.022	0.051	-1.016	9.968	1.969	0.036	-0.009	0.037	0.030
2	9.717	2.080	-0.047	0.044	-0.048	-0.001	9.702	-0.009	2.034	0.029	-0.030	-0.010
3	8.505	-0.057	-0.967	-0.077	-0.012	0.030	8.333	-0.010	-0.959	2.322	-0.031	0.010
4	7.492	-0.021	-0.025	2.099	-0.055	-1.027	9.501	-0.008	0.016	-0.030	2.058	0.039
5	9.561	-0.010	0.033	0.064	2.061	-0.112	9.654	0.003	0.041	0.050	-0.040	2.001

(c) individual population: 108

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	3.620	-0.240	0.060	1.061	0.046	-1.053	8.603	2.299	0.064	-0.075	0.039	0.175
2	8.430	2.001	-0.050	0.238	0.075	-0.390	8.752	-0.161	2.300	0.239	0.000	-0.035
3	10.100	0.022	-1.009	0.035	-0.030	-0.024	10.047	0.029	-1.022	2.119	-0.085	-0.017
4	7.738	-0.021	-0.039	2.115	-0.044	-1.012	9.822	-0.012	0.017	0.023	2.000	0.027
5	9.171	0.014	0.014	0.050	2.100	-0.013	9.261	0.025	0.020	0.029	-0.059	2.050

(d) individual population: 144

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	4.654	-0.064	-0.011	1.020	-0.003	-1.000	9.622	2.011	0.024	-0.007	-0.000	0.050
2	9.950	2.060	-0.029	-0.013	0.000	0.006	9.960	0.008	2.004	-0.005	0.000	0.005
3	8.614	-0.037	-0.934	-0.065	-0.011	0.000	8.504	-0.057	-0.932	2.300	-0.027	-0.009
4	9.801	0.000	0.000	1.866	0.104	-0.840	11.881	0.011	0.000	0.135	1.808	0.010
5	9.158	0.003	0.023	0.054	2.091	-0.159	9.235	0.010	0.032	0.024	-0.054	2.010

(e) individual population: 180

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	5.000	-0.026	0.006	0.969	0.040	-0.968	9.999	1.959	0.010	-0.022	0.033	0.044
2	10.016	2.000	0.001	0.003	0.026	-0.002	9.990	-0.001	1.998	-0.003	0.019	0.000
3	9.494	-0.018	-1.036	-0.021	-0.014	0.013	9.420	-0.035	-1.039	2.143	-0.047	0.010
4	11.000	0.280	-1.000	1.739	0.120	-0.800	12.055	0.224	-1.0194	0.054	1.850	0.022
5	9.210	0.010	0.011	0.040	2.089	-0.119	9.273	0.014	0.015	0.017	-0.051	2.059

(f) individual population: 360

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	5.345	0.005	0.014	0.972	0.033	-0.995	10.312	1.860	0.039	0.000	0.029	-0.033
2	9.850	2.000	-0.008	-0.025	0.010	0.001	9.876	-0.005	2.002	-0.016	0.004	0.008
3	7.907	0.000	-1.000	-0.150	0.000	-0.007	7.861	0.018	-1.016	2.400	-0.010	0.000
4	7.946	0.000	-0.012	1.995	0.000	-0.999	9.968	0.000	-0.003	-0.002	2.000	0.011
5	9.455	-0.020	0.045	0.066	2.084	-0.124	9.598	0.007	0.052	0.040	-0.035	2.008

達する世代数はより大きくなる傾向にあることが分かる。同様に、同図 (e) において、個体数が少ないものは早く収束し、個体数が多くなるほど収束が遅くなるが、世代数が大きいところでも適応度が改善される傾向にあることが見てとれる。さらに、個体数と最終適

応度の関係を表した同図 (f) は、このような変化の状態を直接的に、より明確に示している。

一方、それぞれの場合に得られた S-system パラメータの推定結果の一部を表 10 に示す。同表より、個体数が多いほど目標値に近い値を得る傾向があり、遺伝

子間の相互作用の状態をより明確に表すものと思われる．これを定量的に表すため，先述した相対誤差の絶対値の総和 σ を求めた結果を図 15 に示した．なお，図中の直線は最小二乗法により求めたものである．これより，個体数が多いほど，より目標値に近い値を得る傾向があることが分かる．この結果は，図 14 (f) の適応度の変化と同じ傾向を示しており，本質的には個体数が多くなるにつれて最適化が進むことを示唆する．

これに関連して，前述の GP を用いた他手法²⁾による実験で，個体数を多くすることによって高い正解率を得る一方で，正解を獲得するまでの評価回数が多くなるかかると示されているが，本手法での実験結果もこれを支持するものとなっている．

このように個体数を多くすることによって，比較的良好な結果が得られたのは，個体の多様性を保つことができたことに起因するものと考えられる．そしてその結果，より良好な探索過程を実現でき，目標値により近いパラメータ値が得られたものと思われる．

4.3.2.4 大規模遺伝子ネットワークへの拡張

より規模の大きな遺伝子ネットワークのパラメータ推定に本手法を拡張するため，30 個の遺伝子からなる遺伝子ネットワークモデルに適用することを試みた．なお，実験条件は，(1) 探索領域： α_i と β_i が 0.0 ~ 3.0， $g_{i,j}$ と $h_{i,j}$ は -3.0 ~ 3.0，(2) データセット数：20 セット，(3) 個体数：124 個 (1 分割問題のパラメータ数 62 個の 2 倍) とし，その他は 5 ノード遺伝子ネットワークの場合と同じとした．

この 30 ノード遺伝子ネットワークのターゲットモデルを図 16 に示す．このモデルは，筆者らが前回，報

告したときの例¹⁸⁾と同じものである．目標パラメータ値を表 11 に示す．これに基づいて，上述と同様に，あらかじめ時系列データを計算して求め，これを用いて S-system の各パラメータを推定するものとした．

まず，大規模遺伝子ネットワークへの拡張の可能性を調べるとともに，30 ノード遺伝子ネットワークのパラメータ推定に必要なとされる計算量の目安を得るために，推定するパラメータ数を限定して数値実験を行った．手続きとしては，0 の値となるパラメータはあらかじめ 0 の値で固定し，求めるパラメータ数を 128 個とした．

数値実験の結果として，図 17 に 30 遺伝子各々についての適応度の推移を示す．同図 (a) はノイズを含まない場合，(b) は 10% のガウシアンノイズを含んだ場合をそれぞれ示す．ここで，世代は上述の 5 ノード遺伝子ネットワークの場合と同様に，1 交叉を 1 世代とした．そして，1 パラメータあたり平均 100 万回 (1 分割問題あたり 300 万 ~ 600 万回) の交叉が行われるように設定した．この結果は，上述の 5 ノード遺伝子ネットワークの場合と同様，世代経過とともに適応度が改善されていくことを示している．また，10% のガウシアンノイズを含んだ方は，ノイズの影響を受けて適応度の改善率は低減している．

なお，図 17 (a) は，後述する全パラメータの推定の場合 (図 18 (a)) と比較しても，適応度推移曲線が分散している．これは，多数のパラメータが既知の値であることと，残りの未知のパラメータ数も 1 分割問題あたり 3 ~ 6 個と異なるためで，急速に減少した後，最

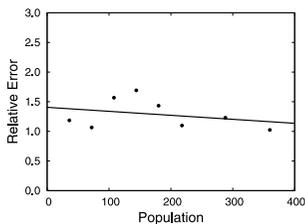


図 15 個体数による目標値との相対誤差の変化

Fig. 15 Variation of relative error of parameters to target values by individual population.

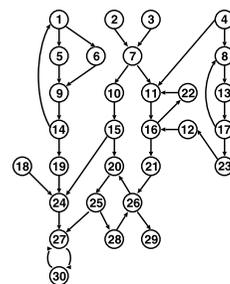


図 16 30 ノード遺伝子ネットワークモデル

Fig. 16 Genetic network model of 30-nodes.

表 11 目標モデル (30 ノード遺伝子ネットワーク) の S-system パラメータ
Table 11 S-system parameters of target model (30-nodes genetic network).

α_i	1.0
β_i	1.0
$g_{i,j}$	$g_{1,14} = -0.1, g_{5,1} = 1.0, g_{6,1} = 1.0, g_{7,2} = 0.5, g_{7,3} = 0.4, g_{8,4} = 0.2, g_{8,17} = -0.2, g_{9,5} = 1.0, g_{9,6} = -0.1, g_{10,7} = 0.3, g_{11,4} = 0.4, g_{11,7} = -0.2, g_{11,22} = 0.4, g_{12,23} = 0.1, g_{13,8} = 0.6, g_{14,9} = 1.0, g_{15,10} = 0.2, g_{16,11} = 0.5, g_{16,12} = -0.2, g_{17,13} = 0.5, g_{19,14} = 0.1, g_{20,15} = 0.7, g_{20,26} = 0.3, g_{21,16} = 0.6, g_{22,16} = 0.5, g_{23,17} = 0.2, g_{24,15} = -0.2, g_{24,18} = -0.1, g_{24,19} = 0.3, g_{25,20} = 0.4, g_{26,21} = -0.2, g_{26,28} = 0.1, g_{27,24} = 0.6, g_{27,25} = 0.3, g_{27,30} = -0.2, g_{28,25} = 0.5, g_{29,26} = 0.4, g_{30,27} = 0.6, other g_{i,j} = 0.0$
$h_{i,j}$	1.0, if $i = j$, 0.0, otherwise

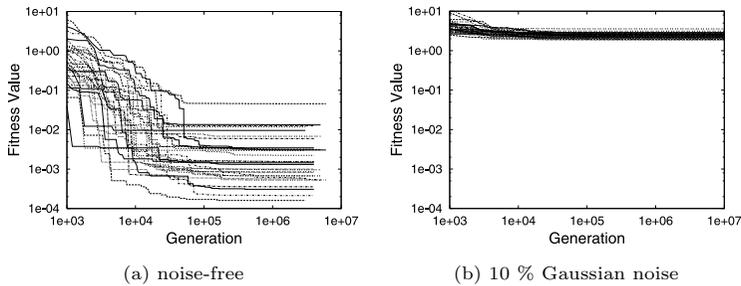


図 17 30 ノード遺伝子ネットワークの適応度の推移 (パラメータ限定)

Fig. 17 Transitions of fitness in 30-nodes genetic network (restricted parameters).

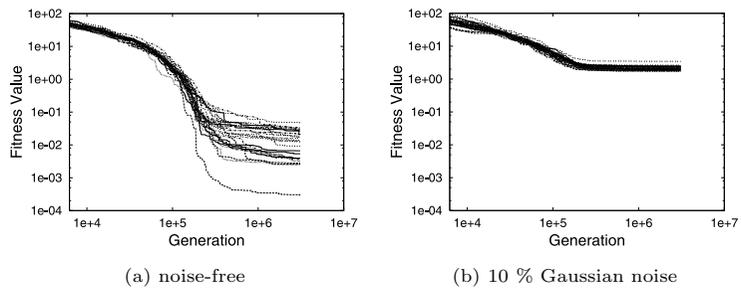


図 18 30 ノード遺伝子ネットワークの適応度の推移 (全パラメータ)

Fig. 18 Transition of fitness in 30-nodes genetic network (all parameters).

表 12 数値実験により得られた 30 ノード遺伝子ネットワークの S-system パラメータ (パラメータ限定)

Table 12 Estimated S-system parameters for 30-nodes genetic network model (restricted parameters).

α_i	$\alpha_1 = 0.990, \alpha_2 = 0.995, \alpha_3 = 0.991, \alpha_4 = 0.998, \alpha_5 = 1.000, \alpha_6 = 0.997, \alpha_7 = 0.994, \alpha_8 = 0.999, \alpha_9 = 0.996, \alpha_{10} = 0.999, \alpha_{11} = 0.990, \alpha_{12} = 0.997, \alpha_{13} = 1.005, \alpha_{14} = 0.999, \alpha_{15} = 1.001, \alpha_{16} = 1.002, \alpha_{17} = 0.988, \alpha_{18} = 0.997, \alpha_{19} = 0.999, \alpha_{20} = 0.988, \alpha_{21} = 0.997, \alpha_{22} = 1.001, \alpha_{23} = 0.999, \alpha_{24} = 0.990, \alpha_{25} = 0.999, \alpha_{26} = 1.004, \alpha_{27} = 1.003, \alpha_{28} = 0.997, \alpha_{29} = 0.995, \alpha_{30} = 0.998$
β_i	$\beta_1 = 0.989, \beta_2 = 0.994, \beta_3 = 0.990, \beta_4 = 0.998, \beta_5 = 0.999, \beta_6 = 0.996, \beta_7 = 0.993, \beta_8 = 0.999, \beta_9 = 0.996, \beta_{10} = 0.999, \beta_{11} = 0.988, \beta_{12} = 0.996, \beta_{13} = 1.004, \beta_{14} = 0.998, \beta_{15} = 1.001, \beta_{16} = 1.000, \beta_{17} = 0.988, \beta_{18} = 0.997, \beta_{19} = 0.999, \beta_{20} = 0.988, \beta_{21} = 0.996, \beta_{22} = 1.001, \beta_{23} = 0.999, \beta_{24} = 0.998, \beta_{25} = 0.998, \beta_{26} = 1.005, \beta_{27} = 1.003, \beta_{28} = 0.996, \beta_{29} = 0.994, \beta_{30} = 0.999$
$g_{i,j}$	$g_{1,14} = -0.099, g_{5,1} = 0.977, g_{6,1} = 0.977, g_{7,2} = 0.487, g_{7,3} = 0.389, g_{8,4} = 0.193, g_{8,17} = -0.195, g_{9,5} = 0.994, g_{9,6} = -0.091, g_{10,7} = 0.293, g_{11,4} = 0.390, g_{11,7} = -0.202, g_{11,22} = 0.394, g_{12,23} = 0.095, g_{13,8} = 0.582, g_{14,9} = 0.992, g_{15,10} = 0.194, g_{16,11} = 0.489, g_{16,12} = -0.192, g_{17,13} = 0.498, g_{19,14} = 0.097, g_{20,15} = 0.689, g_{20,26} = 0.294, g_{21,16} = 0.589, g_{22,16} = 0.488, g_{23,17} = 0.196, g_{24,15} = -0.194, g_{24,18} = -0.097, g_{24,19} = 0.291, g_{25,20} = 0.390, g_{26,21} = -0.193, g_{26,28} = 0.094, g_{27,24} = 0.585, g_{27,25} = 0.293, g_{27,30} = -0.192, g_{28,25} = 0.486, g_{29,26} = 0.390, g_{30,27} = 0.583, \text{other } g_{i,j} = 0.0$
$h_{i,j}$	$h_{1,1} = 1.001, h_{2,2} = 0.997, h_{3,3} = 1.004, h_{4,4} = 0.995, h_{5,5} = 0.999, h_{6,6} = 1.000, h_{7,7} = 1.000, h_{8,8} = 0.993, h_{9,9} = 1.006, h_{10,10} = 0.997, h_{11,11} = 1.003, h_{12,12} = 0.996, h_{13,13} = 0.993, h_{14,14} = 0.997, h_{15,15} = 0.991, h_{16,16} = 1.002, h_{17,17} = 1.012, h_{18,18} = 0.994, h_{19,19} = 0.990, h_{20,20} = 1.010, h_{21,21} = 0.998, h_{22,22} = 0.992, h_{23,23} = 0.994, h_{24,24} = 1.003, h_{25,25} = 0.996, h_{26,26} = 0.990, h_{27,27} = 0.994, h_{28,28} = 1.001, h_{29,29} = 0.998, h_{30,30} = 0.991, 0.0, \text{otherwise}$

終段階での収束速度の違いによるものと考えられる。

一方、収束までの計算量に関しては、どの遺伝子も 1 分割問題あたり交叉数が 10 万回前後 (1 パラメータあたり 3 万回程度) で適応度は一定と見なせる値にまでなり、この段階でほぼ収束したものと考えられる。

この実験によって得られた各パラメータ値をそれぞれ、表 12 と表 13 に示す。同表に見られるように、ノイズのない場合では、各パラメータ値は目標値 (表 11) にほぼ等しい値となっている。また、ノイズのある場合はノイズの影響を受けて目標値と相異なるものもあるが、遺伝子間の相互作用の存在が認識できる程度の値は得られるものと思われる。

次に、30 ノード遺伝子ネットワークのすべてのパ

ラメータ (1 分割問題あたり 62 個、全部で 1,860 個) を数値実験により推定する。交叉回数としては、上述の推定数を限定した場合の数値実験の結果に基づき、収束領域に達すると考えられる、1 パラメータあたり平均 8 万回 (1 分割問題あたり 500 万回) に設定した。

数値実験の結果として、全遺伝子についての適応度の推移を図 18 に示す。なお、同図 (a) はノイズを含まない場合、(b) は 10% のガウシアンノイズを含んだ場合の結果である。同図において、程度の差はあっても、いずれも適応度は収束していることが観察できる。

この実験によって得られた各パラメータ値をそれぞれ、表 14、表 15 に示す。なお、両表には、得られたパラメータのうち、 $g_{i,j}$ と $h_{i,j}$ についてはその絶対値

表 13 ノイズデータを用いた数値実験により得られた 30 ノード遺伝子ネットワークの S-system パラメータ (パラメータ限定)

Table 13 Estimated S-system parameters for 30-nodes genetic network model with noisy data (restricted parameters).

Table with 4 columns: parameter label (alpha_i, beta_i, g_i,j, h_i,j), parameter name, and numerical value. Includes parameters like alpha_1 to alpha_10, beta_1 to beta_10, g_1,14 to g_9,9, and h_1,1 to h_26,26.

表 14 数値実験により得られた 30 ノード遺伝子ネットワークの S-system パラメータ (全パラメータ)

Table 14 Estimated S-system parameters for 30-nodes genetic network model (all parameters).

Table with 4 columns: parameter label, parameter name, and numerical value. Includes parameters like alpha_1 to alpha_30, beta_1 to beta_30, g_1,1 to g_30,30, and h_1,1 to h_30,30.

表 15 ノイズデータを用いた数値実験により得られた 30 ノード遺伝子ネットワークの S-system パラメータ (全パラメータ)

Table 15 Estimated S-system parameters for 30-nodes genetic network model with noisy data (all parameters).

Table with 4 columns: parameter label, parameter name, and numerical value. Includes parameters like alpha_1 to alpha_30, beta_1 to beta_30, g_1,1 to g_30,30, and h_1,1 to h_30,30.

が各遺伝子中上位 5 位までのものを記述している。ただし、その範囲内でも値が 0 のときは記載していない。

同表より、ノイズのある場合とない場合のパラメータ値を比較すると、明らかにノイズのある場合はターゲットモデルで設定した目標値からの差異が大きくなっていることが分かる。これをより定量的に表すため、目標パラメータ値と絶対誤差の絶対値の和および相対誤差の絶対値の和をそれぞれ計算し、表 16 に

示した。この表から、実験値は目標値から離れていることが分かる。なお、同表には前報の GA を用いた場合(18)の値もあわせて示しているが、これより両手法による結果はほぼ同様な傾向があることが分かる。

この実験によって、(1)ノイズのある場合でも、ターゲットモデルで設定した目標値と正確な値までは一致め、目標遺伝子間の相互作用を示す主要なパラメータはほぼ導出できている。(2) ターゲットモデルで 0 と

表 16 目標パラメータとの差異
Table 16 Differences from target parameters.

手法	絶対誤差	相対誤差
GP (noise free)	107.0	37.7
GP (10 % noise)	159.4	98.7
GA (noise free)	44.2	11.3
GA (10 % noise)	155.2	69.9

設定したパラメータは 0 あるいはほぼ 0 に近い値が得られている、そして上述の (3) 今回得られたパラメータ値は、GA を用いた場合の値¹⁸⁾ に比べてやや精度は劣るが大きな差はなくほぼ同程度である、などの知見を得た。このことから、本実験結果は生物学的ネットワークを解釈するには有効であるものと思われる。

以上のことから、本提案手法は、基本的な小規模遺伝子ネットワークに合わせて、より規模の大きな遺伝子ネットワークに対しても S-system パラメータの推定に適用可能であるものと考えられる。なお、精度を高めるための最適化技術については今後の課題となる。

5. おわりに

本研究では、遺伝的プログラミングを用いた関数最適化手法を提案し、ベンチマーク関数に適用してその性能評価を行った。そして、実問題として遺伝子ネットワークの S-system モデルのパラメータの推定を行い、その有効性を検討した。

ベンチマーク関数に適用した結果から、本手法は、良好な解が得られることが示された。特に関数最適化問題において困難とされる性質の 1 つである多峰性を有する関数に対して局所解に陥らず、最適解に達することが分かった。

次に、遺伝子ネットワークの S-system パラメータの推定に関して、5 つの遺伝子を有する遺伝子ネットワークモデルに適用した結果、十分正確なパラメータ値を得ることができ、実際の遺伝子ネットワーク推定の適用において有効であるものと思われた。

さらに、より規模の大きな遺伝子ネットワークのパラメータ推定への拡張性を確かめるため、30 ノード遺伝子ネットワークモデルに適用することを試みた。その結果、ターゲットモデルの持つ遺伝子間の相互作用の状態を示すパラメータをほぼ導出できたことから、規模の大きな遺伝子ネットワークに対しても拡張可能であるものと考えられた。

なお、現在、大規模遺伝子ネットワーク推定の実用化に向けて、より高速に解探索過程を実現しうる GP の適応技術の開発を目的として、複数の計算を同時に効率良く実行することのできる手法について検討している。これについては今後の課題としたい。

参考文献

- 1) 小野 功, 山村雅幸, 喜多 一: 実数値 GA とその応用, 人工知能学会誌, Vol.15, No.2, pp.259-266 (2000).
- 2) 杉本直也, 坂本栄理奈, 伊庭斉志: 遺伝的プログラミングによる微分方程式の推定, 人工知能学会論文誌, Vol.19, No.6A, pp.450-459 (2004).
- 3) 木村周平, 高橋 治, 小林重信, 小長谷明彦: 並列化に適した遺伝的ローカルサーチによる非線形関数最適化, 計測自動制御学会論文誌, Vol.40, No.4, pp.448-457 (2004).
- 4) 樋口隆秀, 筒井茂義, 山村雅幸: 実数値 GA におけるシンプレックス交叉の提案, 人工知能学会論文誌, Vol.16, No.1, pp.147-155 (2001).
- 5) 廣安知之, 三木光範, 佐野正樹, 谷村勇輔, 濱崎雅弘: 2 個体分散遺伝的アルゴリズム, 計測自動制御学会論文誌, Vol.38, No.11, pp.990-995 (2002).
- 6) Koza J.: *Genetic Programming*, MIT Press (1992).
- 7) 松村幸輝, 村井保之: 遺伝的プログラミングに基づくファジイロボットの障害物回避モデル, 電子情報通信学会論文誌, Vol.J83-A, No.12, pp.1539-1551 (2000).
- 8) 松村幸輝, 後藤道彦, 浜松芳夫: 遺伝的プログラミングによるチューニング手法を用いたファジイ交渉エージェント, 電気学会論文誌, Vol.121-C, No.7, pp.1160-1170 (2001).
- 9) 松村幸輝: Walsh 変換を用いた遺伝的プログラミング手法に基づくファジイ交渉エージェント, 情報処理学会論文誌, Vol.46, No.1, pp.1203-1214 (2005).
- 10) 伊庭斉志: 遺伝的プログラミングと進化論的計算手法, 人工知能学会誌, Vol.15, No.2, pp.215-258 (2000).
- 11) 木村周平, 小長谷明彦: 距離に依存せずに多様性を制御する GA による高次元関数最適化, 人工知能学会論文誌, Vol.18, No.4, pp.193-202 (2003).
- 12) 喜多 一, 小野 功, 小林重信: 実数値 GA のための正規分布交叉の多数の親を用いた拡張法の提案, 計測自動制御学会論文誌, Vol.36, No.10, pp.875-883 (2000).
- 13) Takahashi, O., Kita, H. and Kobayashi, S.: A Distance Dependent Alternation Model on Real-coded Genetic Algorithm, *IEEE Int. Conf. on SMC'99*, pp.619-624 (1999).
- 14) Sakuma, J. and Kobayashi, S.: Extrapolation-Directed Crossover for Real-coded GA, Overcoming Deceptive Phenomena by Extrapolative Search, *Proc. CEC*, pp.655-662 (2001).
- 15) 佐藤 浩, 小野 功, 小林重信: 遺伝的アルゴリズムにおける世代交代モデルの提案と評価, 人工知能学会論文誌, Vol.12, No.5, pp.734-744 (1997).

- 16) Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K. and Tomita, M.: Dynamic modeling of genetic networks using genetic algorithm and S-system, *Bioinformatics*, Vol.19, pp.643–650 (2003).
- 17) Maki, Y., Ueda, T., Okamoto, M., Uematsu, N., Inamura, Y. and Eguchi, Y.: Inference of Genetic Networks Using the Expression Profile Time Course Data of Mouse P19 Cells, *Genome Informatics*, Vol.13, pp.382–383 (2003).
- 18) Kimura, S., Hatakeyama, M. and Konagaya, A.: Inference of S-system Models of Genetic Networks from Noisy Time-series Data, *ChemBio Informatics J.*, Vol.4, No.1, pp.1–14 (2004).
- 19) Press, W., Teukolsky, S., Vetterling, W. and Flannery, B.: *Numerical Recipes in C*, 2nd edition, Cambridge University Press (1995).
- 20) Cleveland, W.S.: Robust Local Weight Regression and Smoothing Scatterplots, *J. American Statistical Association*, Vol.79, pp.829–836 (1979).
- 21) Tominaga, D., Koga, N. and Okamoto, M.: Efficient Numerical Optimization Algorithm Based on Genetic Algorithm for Inverse Problem, *Proc. Genetic and Evolutionary Computation Conference 2000*, pp.251–258 (2000).
- 22) Ando, S., Sakamoto, E., and Iba, H.: Evolutionary modeling and inference of gene network, *Information Sciences*, Vol.145, No.3–4, pp.237–29 (2002).

(平成 17 年 3 月 26 日受付)

(平成 17 年 9 月 2 日採録)



松村 幸輝 (正会員)

1978 年大阪市立大学大学院博士課程修了。茨城大学工学部助教授，甲南大学理学部教授を経て，2002 年より鳥取大学工学部教授・工学博士。主として，進化計算，知識工学，知能情報科学等の研究に従事。電子情報通信学会，電気学会，人工知能学会，日本知能情報ファジイ学会等会員。



種田 洋

2005 年鳥取大学工学部知能情報工学科卒業。現在，同大学院博士前期課程在学。進化計算，システムモデリングの研究に従事。



木村 周平

2001 年東京工業大学大学院博士後期課程修了。理化学研究所ゲノム科学総合研究センター研究員を経て，2004 年より鳥取大学工学部助教授・博士（工学）。進化計算，バイオインフォマティクス等の研究に従事。人工知能学会，日本分子生物学会，日本認知科学学会等会員。