

Product unit based neural networks を用いた 遺伝子ネットワークの S-system モデル推定手法の提案

村田 裕章^{†1} 越野 亮^{‡2}
三田村 公智^{†1} 木村 春彦^{†1}

本研究ではニューラルネットワークの一種である Product Unit based Neural Network (PUNN) を用いて、遺伝子間の制御関係を表現する遺伝子ネットワークを推定する手法を提案する。提案手法では微分方程式を解くのではなく、PUNN を学習させることで、微分方程式で記述される S-system モデルの遺伝子ネットワークの推定を行う。評価実験の結果、S-system モデルの遺伝子ネットワークを推定する従来手法に比べ、同程度の性能を保持しつつも、約 160 倍の高速化を実現できたことを示す。

Inference of S-system Models of the Genetic Networks Using the Product-unit-based-neural-networks

HIROAKI MURATA,^{†1} MAKOTO KOSHINO,^{‡2}
MASATOMO MITAMURA^{†1} and HARUHIKO KIMURA^{†1}

In this study, we proposed the method of inference of genetic networks which expresses the regulation of genes. The proposed method does not solve the differential equations, learns the genetic networks using Product-Unit-based-Neural-Network (PUNN) and estimates the S-system model of genetic networks which describes the differential equations. The experimental results show the proposal method is 160 times faster than the previous method which estimated S-system model of genetic networks while maintaining equivalent performance to the previous method.

^{†1} 金沢大学

Kanazawa University

^{‡2} 石川工業高等専門学校

Ishikawa National College of Technology

1. はじめに

生体内では様々な遺伝子が互いに制御し合い、発現量を調節することで生命活動の維持が行われている。そのような制御関係は遺伝子ネットワークと呼ばれ、新薬の開発や病気の治療に応用できることから、遺伝子ネットワークの解析や推定を行う研究が行われている^{1),2)}。特に DNA マイクロアレイ技術の進歩により、細胞全体レベルでの遺伝子発現量の解析が可能になってきており、これらの情報を用いた遺伝子ネットワーク推定に注目が集まっている。

遺伝子発現量の時系列データから、遺伝子間の制御関係(遺伝子ネットワーク)を推定することは、遺伝子ネットワーク推定問題と呼ばれ、与えられた遺伝子発現量の時系列データを再現する数理モデルを求めることが目的となる。遺伝子ネットワークを示す数理モデルとして、ブリアンネットワークやベイジアンネットワーク、微分方程式モデルなどが提案されている^{1),2)}。それらのモデルの中で本研究では、微分方程式モデルの一種である S-system モデル³⁾ とニューラルネットワークモデル(NN モデル)⁴⁾ に注目する。

S-system モデルは環境や未知の転写制御因子など、多くの要素を省略したモデルであるが、遺伝子の制御関係について有意義なヒントを得る手法として期待されている。しかしながらネットワーク推定の際に、微分方程式を繰り返し解く必要があり膨大な計算量を必要とする。なお、一般に与えられる遺伝子発現量の時系列データには観測誤差が多く含まれることから、S-system モデルを用いて遺伝子ネットワーク推定を行う際には、遺伝子ネットワークの大まかな構造を求めようとする研究も多く行われているが、本研究では与えられた遺伝子発現量の時系列データを正確に再現することを目標とする。NN モデルは、微分方程式を遺伝子発現量と発現変化量の関係式と見なし、階層型ニューラルネットワーク(階層型 NN)の学習により、その近似関数を求めることで遺伝子ネットワーク推定を行う。これにより微分方程式を解く必要がなくなり、大幅な計算量削減に成功している。しかしながら微分方程式の近似関数を階層型 NN の入出力関係として求めているために、近似関数がブラックボックス化しており、どの遺伝子がどの遺伝子をどのような形で制御しているのかという遺伝子間の制御関係の詳細を得ることは難しい。そのため NN の結合荷重に対して感度解析を行い遺伝子間の制御関係の抽出が行われているが、関係性の有無および正負の 3 種類の関係しか特定されておらず、遺伝子間の制御関係について十分な情報を得られるとはいえない。

そこで本研究では NN モデルにおいて階層型 NN ではなく、Product Unit based NN (PUNN)⁵⁾ を用いる PUNN モデルを考え、PUNN モデルを用いた遺伝子ネットワーク推定

手法を提案する．PUNN モデルを用いた推定手法では遺伝子発現量と発現変化量の関係を PUNN の学習によって得るため，NN モデルを用いた推定手法と同様に微分方程式を解く必要がなく，S-system モデルを用いた推定手法に比べ大幅な計算量の削減を期待できる．また PUNN では，入力を遺伝子発現量，出力を発現変化量とすることで入出力関係を S-system モデルと同じ微分方程式で定義できる．そのため PUNN モデルでは結合荷重と S-system モデルの間に明確な対応関係があり，S-system モデルと同程度の詳細な遺伝子間の制御関係を求めることができる．評価実験の結果，S-system モデルを用いた推定手法と比べ得られた最良パラメータは良くならなかったが，得られるパラメータの平均値は S-system モデルを用いた推定手法よりも良くなり，実行速度に関しては約 160 倍の速度向上を実現できたことを示す．また PUNN モデルを用いた推定手法において，与える発現変化量に含まれる誤差が少ないほど最良値，平均値ともに良いパラメータが求められ，誤差が含まれない正確な発現変化量が与えられると仮定した場合，誤差 0 のパラメータを求められたことも示す．

2. 従来研究：微分方程式モデルを用いた遺伝子ネットワーク推定方法

2.1 微分方程式モデル

微分方程式モデルを用いると， N 個の遺伝子からなる遺伝子ネットワークは式 (1) で表現される．

$$\frac{dX_i}{dt} = G_i(X_1, \dots, X_N) \quad (i = 1, \dots, N) \quad (1)$$

ここで X_i は遺伝子 i の発現量， $\frac{dX_i}{dt}$ は遺伝子 i の発現変化量（転写速度に相当）を示す． G_i は $\frac{dX_i}{dt}$ を定義する関数であり，微分方程式モデルを用いる遺伝子ネットワーク推定において，この G_i を求めることが目的となる．しかしながら一般形で表現される G_i を求めることは難しく，線形関数などの仮定を設けることで G_i を求めることが多く，微分方程式モデルにおいても様々なモデルが提案されている．本研究ではそれらのモデルの中で S-system モデル³⁾ と NN モデル⁴⁾ に注目する．

2.2 S-system モデルを用いた推定手法

2.2.1 S-system モデル

S-system モデル³⁾ は環境や未知の転写制御因子など，多くの要素を省略したモデルであるが，遺伝子の制御関係について有意義なヒントを得る手法として期待されている．S-system モデルはすべての遺伝子についてその関与を意味するパラメータを含むモデルであり，遺伝子 i の発現量を X_i で表すと，関数 G_i ($i = 1, \dots, N$) を次の関数で表現している．

$$\begin{cases} \frac{dX_1}{dt} = \alpha_1 \prod_{j=1}^N X_j^{g_{1,j}} - \beta_1 \prod_{j=1}^N X_j^{h_{1,j}} \\ \vdots \\ \frac{dX_N}{dt} = \alpha_N \prod_{j=1}^N X_j^{g_{N,j}} - \beta_N \prod_{j=1}^N X_j^{h_{N,j}} \end{cases} \quad (2)$$

なお $g_{i,j}$ は X_i の発現過程に關与する X_j の相互作用係数， $h_{i,j}$ は X_i の抑制過程に關与する X_j の相互作用係数， α_i ， β_i はそれぞれ X_i の発現項，抑制項に乗じる係数である．このモデルは複数の発現・抑制過程からなる反応経路を近似した表現となっており，詳細な構造が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⁶⁾．S-system モデルを用いることで， N 個の遺伝子からなる遺伝子ネットワークを $2N(N+1)$ 個の S-system パラメータ α_i ， β_i ， $g_{i,j}$ ， $h_{i,j}$ で表現できる．また $g_{i,j}$ と $h_{i,j}$ は遺伝子 i と遺伝子 j 間の制御関係を表し，これらの値によって遺伝子ネットワークが大きく変化することから，キネティックオーダー (kinetic order) と呼ばれている^{6),7),*1}．

2.2.2 S-system モデルを用いた推定手法

S-system モデルを用いた遺伝子ネットワーク推定問題を最適化問題として考えた場合，与えられた遺伝子発現量の時系列データを再現する S-system パラメータを求める問題として，以下のように定式化される．

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

ここで $X_{exp,i,t}$ は実験的に得られた時刻 t における遺伝子 i の発現量， $X_{cal,i,t}$ は連立微分方程式 (2) を解くことによって得られる遺伝子発現量， N はネットワークに含まれる遺伝子数， T は実験的に得られた時系列データのタイムポイント数である．

しかしながら遺伝子ネットワーク推定問題の問題空間は $2N(N+1)$ 次元であるため，既存の最適化手法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困難である．そのため問題をいくつかのサブ問題（各遺伝子に相当）に分割し，遺伝子ネットワークの推定を行う方法が提案されている⁶⁾．遺伝子 i に相当するサブ問題は次のように定式化される．

*1 正確にはキネティックオーダーとは化学反応論/反応速度論における反応次数を指すため， $g_{i,j}$ や $h_{i,j}$ はキネティックオーダーに相当する量となるが，本研究では文献 6)，7) に倣いキネティックオーダーと呼ぶ．

$$f_i = \sum_{t=1}^T \left(\frac{X'_{cal,i,t} - X_{exp,i,t}}{X_{exp,i,t}} \right)^2 \quad (4)$$

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^N Y_j^{g_{i,j}} - \beta_i \prod_{j=1}^N Y_j^{h_{i,j}} \quad (5)$$

$$Y_j = \begin{cases} X_j^c & \text{if } j = i \\ \hat{X}_j & \text{if } j \neq i \end{cases} \quad (6)$$

ここで $X'_{cal,i,t}$ は微分方程式 (5) を解くことで得られる時刻 t における遺伝子 i の発現量を示し、 \hat{X}_j は実験的に得られた時系列データから補間することで求める。補間方法として与えられた遺伝子発現量の時系列データに実験誤差などのノイズが含まれない場合はスプライン補間を用い、ノイズが含まれると考えられる場合は局所線形回帰などの平滑化手法を用いて補間を行う⁶⁾。このようにサブ問題に分割することで、 $2N(N+1)$ 次元の遺伝子ネットワーク推定問題を、 N 個の $2(N+1)$ 次元のサブ問題に変換できる。

S-system モデルは自由度が高いため複数の解が存在する可能性があり、また一般的に与えられる遺伝子発現量の時系列データには実験誤差が含まれるため、正確な S-system パラメータを求めることは困難である。そこで遺伝子ネットワークの結合は疎であるという特徴を導入した目的関数が提案されている^{8),*1}。S-system モデルにおいて $g_{i,j}$ は X_i の発現過程に参与する X_j の相互作用係数であり、 $g_{i,j} = 0$ とは遺伝子 j は遺伝子 i の発現過程に参与していないことを意味する。同様に $h_{i,j}$ は抑制過程の相互作用係数であり、 $h_{i,j} = 0$ とは遺伝子 j は遺伝子 i の抑制過程に参与していないことを意味する。このことから遺伝子ネットワークの結合が疎になる場合、多くのキネティックオーダー ($g_{i,j}$ や $h_{i,j}$) が 0 になることが分かり、以下のペナルティ項が提案されている。

$$P_i = P_c \sum_{j=1}^{2N-I} (|K_{i,j}|) \quad (7)$$

ここで $K_{i,j}$ は遺伝子 i のキネティックオーダーの絶対値を昇順にソートしたときの j 番目の値を意味する。 P_c はペナルティ係数、 N は遺伝子数、 $2N$ はキネティックオーダー数、 I は最大入り次数を示す。ここで、最大入り次数とは遺伝子 i に影響を及ぼす遺伝子の最大数を規

定するパラメータである。また遺伝子ネットワークの特徴の 1 つとして考えられているスケールフリー性を用いる手法⁹⁾も提案されているが、スケールフリー性を有する解候補を生成しており、ペナルティ項として導入されていないため本研究では導入しないこととする。

また統計モデルの良さを評価する際に多く用いられる赤池情報量基準 (Akaike's Information Criteria: AIC) も目的関数に導入されており、Noman らにより以下の目的関数が提案されている⁷⁾。Noman らは実験的に得られた遺伝子発現量 $X_{exp,i,t}$ と推定したパラメータから計算した遺伝子発現量 $X_{cal,i,t}$ との絶対誤差 $\epsilon = (X_{cal,i,t} - X_{exp,i,t})$ が正規分布に従うとし、式 (10) の AIC を導入した目的関数を定義している。ここで Φ はパラメータ数を表す。なお本研究では S-system モデルによる遺伝子ネットワーク推定を行う際には、式 (10) を目的関数とする。

$$\sigma_i^2 = \frac{1}{T} \sum_{k=1}^T (X_{cal,i,t} - X_{exp,i,t})^2 \quad (8)$$

$$\Lambda = -\frac{1}{2\sigma_i^2} (X_{cal,i,t} - X_{exp,i,t})^2 - \frac{T}{2} \ln(2\pi\sigma_i^2) \quad (9)$$

$$f_i^S = -2\Lambda + 2\Phi + P_i \quad (10)$$

2.3 ニューラルネットワークモデル (NN モデル) を用いた推定手法

2.3.1 ニューラルネットワークモデル

微分方程式モデルでは、一般に微分方程式を繰り返し解く必要があるため、多くの計算量を必要とする。そこで式 (1) を微分方程式ではなく、遺伝子発現量 X_1, \dots, X_N とそのときの発現変化量 $\frac{dX_i}{dt}$ の関係を示す関数 G_i ととらえることで、関数近似を行い遺伝子ネットワーク推定を行う手法が提案されている⁴⁾。この手法では関数近似の際に 3 層ニューラルネットワークを使用しており、ニューラルネットワークで遺伝子ネットワークを表現することをニューラルネットワークモデル (NN モデル) と呼んでいる⁴⁾。NN モデルは 1 つの遺伝子発現量の時系列データに注目したモデルであり、 N 個の遺伝子発現量の時系列データに対しては、 N 個の NN モデルが必要となる。NN モデルは、入力層ユニット N 個、中間層ユニット N_H 個 (定数パラメータ)、出力層ユニット 1 個で構成されており、 k 番目の中間層ユニットの出力 y_k^H および出力層ユニットの出力 y^O は、それぞれ式 (11) および式 (12) で計算される。

$$y_k^H = f \left(\sum_{j=1}^N w_{j,k}^{IH} X_j - \theta_k^H \right) \quad (11)$$

*1 後述のローカルサーチでの説明の関係上、文献 7) での表記を用いる。

$$y^O = g \left(\sum_{k=1}^{N_H} w_k^{HO} y_k^H - \theta^O \right) \quad (12)$$

ここで X_j は j 番目の入力層ユニットの入力値, $w_{j,k}^{IH}$ は j 番目の入力層ユニットと k 番目の中間層ユニット間の結合荷重, θ_k^H は k 番目の中間層ユニットの閾値パラメータ, w_k^{HO} は k 番目の中間層ユニットと出力層ユニット間の結合荷重, θ^O は出力層ユニットの閾値パラメータ, $f(x) = (1 + \exp(-x))^{-1}$, $g(x) = x$ を示す. NN モデルにおいて, 入力 X_j を遺伝子 j の遺伝子発現量, 出力 y^O を遺伝子 i の発現変化量に対応させることで, 関数 $G_i(X_1, \dots, X_N)$ を表現する.

2.3.2 ニューラルネットワークモデルを用いた推定手法

NN モデルを用いた遺伝子ネットワーク推定手法では, 入力された遺伝子発現量に対し, その遺伝子の発現変化量を出力するように NN モデルを学習させることで, 関数 $G_i(X_1, \dots, X_N)$ を求める. しかしながらモデルの自由度が高いために与えられた遺伝子発現量データのみで遺伝子ネットワークを推定することは難しい. そのため S-system モデルを用いた推定手法と同様に「遺伝子ネットワークの結合は疎である」という事前知識を導入した式 (13) を用いて NN の学習が行われている. 遺伝子 i が遺伝子 j から制御を受けていない場合, NN モデルでは j 番目の入力層ユニットから中間層ユニットへの結合荷重 $(w_{j,1}^{IH}, \dots, w_{j,N_H}^{IH})$ はすべて 0 となり, それらの結合強度の二乗和 $w_j = \sum_{k=1}^{N_H} (w_{j,k}^{IH})^2$ も 0 となる. そのため結合強度の二乗和 w_j を 0 にすることで, 結合が疎な遺伝子ネットワークを表現できることから, 式 (13) の第 2 項が追加されている. W_j は w_j を昇順にソートしたものの (つまり $W_1 \leq W_2 \leq \dots \leq W_N$ となる), c は定数パラメータ, I は最大入り次数を示す. なお第 2 項は S-system モデルを用いた推定手法で用いられる目的関数におけるペナルティ項に対応している. また式 (13) の第 1 項は NN の学習時によく用いられる誤差関数であり, 入力ベクトル \mathbf{X}_t ($t = 1, \dots, T$) が与えられたときの出力 $y^O(\mathbf{X}_t)$ と教師信号 y_t 間の誤差の総和を表している.

$$E = \frac{1}{2} \sum_{t=1}^T (y^O(\mathbf{X}_t) - y_t)^2 + c \sum_{j=1}^{N-I} W_j \quad (13)$$

しかしながら, NN モデルを用いた推定手法を用いて関数 G_i の近似関数が得られたとしても, NN の入出力関係としてブラックボックス化されているため, どの遺伝子がどの遺伝子を制御しているのかを知ることは難しい. そのため感度解析を用いて結合荷重からそれら

の情報を抽出することで, 関係性の有無および制御の正負の情報を得ている.

3. 提案手法: Product Unit based NN モデル (PUNN モデル) を用いた推定方法

NN モデルを用いた推定手法では, 微分方程式モデルを用いながら微分方程式を解かずに遺伝子ネットワーク推定を行うことで, 大幅な計算量の削減を実現している. しかしながら遺伝子間の制御関係として, 関係性の有無および制御の正負しか得ることができていない. この要因として NN モデルでは, 一般化された微分方程式モデルを基に考えられたために, 詳細な制御関係の記述が定まっていなかったことが大きな要因と考えられる. それ以外にも制御関係を求める際に感度解析を必要としているように, 関数 G_i をブラックボックスとして求めるため, 制御関係と結合荷重間に明確な対応関係がないことも要因の 1 つと考えられる.

そこで本研究では NN モデルを用いた推定手法と同様に, 微分方程式を解かずに遺伝子ネットワーク推定を行いつつ, より詳細な遺伝子間の制御関係を求めることができる Product Unit based Neural Network モデル (PUNN モデル) を用いた推定手法を提案する. PUNN モデルでは詳細な制御関係の記述に S-system モデルを用い, NN の一種である Product Unit based NN (PUNN)⁵⁾ を用いることで, 制御関係 (S-system パラメータ) と結合荷重との間に明確な対応関係を設定できる.

3.1 Product Unit based NN (PUNN)

Product Unit based NN (PUNN)⁵⁾ は入出力関係が総乗で定義される Product Unit を用いた NN であり, 中間層は Product Unit が用いられ, 出力層では入出力関係が総和で定義される一般的なユニット (Summation Unit と呼ぶ) で構成されている. k 番目の中間層ユニットの出力 y_k^H および唯一の出力層ユニットの出力 y^O は, それぞれ式 (14) および式 (15) となる.

$$y_k^H = \prod_{j=1}^N (X_j)^{w_{j,k}^{IH}} \quad (14)$$

$$y^O = \sum_{k=1}^{N_H} w_k^{HO} y_k^H - \theta^O \quad (15)$$

ここで入力層のユニット数を N , 中間層のユニット数を N_H , 出力層のユニット数を 1 とし, X_j は j 番目の入力層ユニットの入力値, $w_{j,k}^{IH}$ は j 番目の入力層ユニットと k 番目の中間層ユニット間の結合荷重, w_k^{HO} は k 番目の中間層ユニットと出力層ユニット間の結合

荷重, θ^O は出力層ユニットの閾値パラメータを示す.

3.2 Product Unit based NN モデル (PUNN モデル)

PUNN を次の設定にすることで, S-system モデルの遺伝子ネットワークを表現する PUNN モデルを定義することができる. 入力層のユニット数を遺伝子数 N , 中間層のユニットを 2 個, 出力層のユニットを 1 個に設定し, j 番目の入力層ユニットと 1 番目の中間層ユニット間の結合荷重 $w_{j,1}^{IH}$ を $g_{i,j}$, j 番目の入力層ユニットと 2 番目の中間層ユニット間の結合荷重 $w_{j,2}^{IH}$ を $h_{i,j}$, 1 番目の中間層ユニットと出力層ユニット間の結合荷重 w_1^{HO} を α_i , 2 番目の中間層ユニットと出力層ユニット間の結合荷重 w_2^{HO} を $-\beta_i$, 出力層のユニットの閾値パラメータ $\theta^O = 0$ とする. この PUNN の入出力関係は, 式 (16) で表され, この式は S-system モデルにおける遺伝子 i の発現量に関する微分方程式 (17) と同じである.

$$y^O = \alpha_i \prod_{j=1}^N X_j^{g_{i,j}} - \beta_i \prod_{j=1}^N X_j^{h_{i,j}} \quad (16)$$

$$\frac{dX_{i,t}}{dt} = \alpha_i \prod_{j=1}^N X_{j,t}^{g_{i,j}} - \beta_i \prod_{j=1}^N X_{j,t}^{h_{i,j}} \quad (17)$$

PUNN モデルの一例として図 1 に 5 つの遺伝子からなる遺伝子ネットワークにおける遺伝子 1 に関する微分方程式を表現した PUNN モデルを示す.

また PUNN モデルの別の見方として, Answer-in-Weights 構造¹⁰⁾ の PUNN と見ることもできる. Answer-in-Weights 構造とは対象とする問題の解が, NN の入出力関係ではなく, 結合荷重ベクトルとして求められる構造である. PUNN モデルにおいて PUNN の結合荷重が

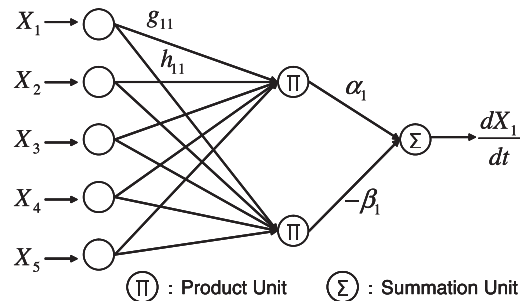


図 1 遺伝子ネットワークの一部を表現した PUNN モデル
Fig. 1 The PUNN model which shows a part of genetic network.

そのまま S-system パラメータを表していることから, PUNN モデルは Answer-in-Weights 構造の PUNN と考えることもできる.

3.3 PUNN モデルを用いた推定手法

PUNN モデルを用いる推定手法は, NN モデルを用いた推定手法と同様に, 入力された遺伝子発現量に対し, その遺伝子の発現変化量を出力するように PUNN モデルを学習することで S-system パラメータを求める. しかしながら, NN の学習手法として一般的な傾き降下法は PUNN の学習に不向きといわれている^{11),12)} ため, 本研究では一般的な最適化手法を用いて学習を行う^{*1}. そのため NN の学習でよく使われる, ネットワークの出力と教師信号間の誤差を表す誤差関数ではなく, 目的関数として式 (18) を定義し, PUNN モデルの学習に用いる. なお式 (18) は入力ベクトル $\mathbf{X}_t = (X_{1,t}, \dots, X_{N,t})$ に対して $y_t (= \frac{dX_{i,t}}{dt})$ を出力する PUNN モデルを学習する際に用いる目的関数を示している.

$$E'_{P,i} = \begin{cases} -\infty & \text{if } e_{P,i} = 0 \\ \log(e_{P,i}) & \text{if } 0 < e_{P,i} < 1 \\ e_{P,i} & \text{if } 1 \leq e_{P,i} \end{cases} \quad (18)$$

$$e_{P,i} = \sum_{t=1}^T (y^O(\mathbf{X}_t) - y_t)^2 \quad (19)$$

式 (19) をそのまま用いない理由は, 式 (19) 上では小さな誤差であっても式 (18) を用いることで大きな誤差となり, より y_t に近い出力を得られるようにするためである.

3.3.1 クラスタリングを用いたペナルティ項

目的関数に「遺伝子ネットワークの結合は疎である」という特徴を導入する方法の 1 つに, キネティックオーダー数を制限するペナルティ項を導入する方法がある. 従来までは, それらの制限を固定 (定数パラメータ) としていた. そこで本研究では制限を動的に決めるペナルティ項を導入する. 基本的な考えは, キネティックオーダーを絶対値を基準に, 2 つのクラスに分け, 絶対値が小さい方のクラスに属するキネティックオーダーを 0 にするようにペナルティを与えるというものである. なお非零のキネティックオーダーのみを対象にクラスタリングを行う. しかし数個のキネティックオーダーの値が 0 となり, それ以外の多くのキネティックオーダーの絶対値が比較的大きな値になる場合, 結合が疎の遺伝子ネットワークを表現していないにもかかわらず, ペナルティ値は小さくなってしまふ. このような事態を避けるため

*1 最適化手法については, 4 章で述べる.

に、値が 0 のキネティックオーダー数が多いほどペナルティ値を小さくするようにする。

クラスターリングを用いてキネティックオーダーを値が 0 であるクラス K_{c0} 、絶対値の小さなクラス K_{c1} 、絶対値の大きなクラス K_{c2} に分け、 K_{nz} を K_{c0} の要素数とすると、クラスターリングを用いたペナルティ項は以下ようになる。

$$P_{P,i} = \begin{cases} \infty & \text{if } K_{nz} < 2 \\ 0 & \text{if } K_{nz} = 2 \\ \sum_{x \in K_{c1}} x + (2N - K_{nz}) & \text{else} \end{cases} \quad (20)$$

N は遺伝子数であり、 $2N$ はキネティックオーダー数となる。非零のキネティックオーダー数が 0 個の場合は遺伝子発現量が一定値となり、1 個の場合は発現過程もしくは抑制過程のみとなり遺伝子発現量が単調増加もしくは単調減少になると考えられる。そのため、発現過程と抑制過程のバランスによって遺伝子発現量が決まるとする S-system モデルを再現するためには、非零のキネティックオーダーが 2 個以上必要であると考えペナルティ項の設計を行った。なお対象とする遺伝子ネットワークによっては、非零のキネティックオーダー数が 2 個以上になる場合もある。そのような場合はペナルティ項が 0 とならない S-system パラメータが最良解になるだけにすぎず、このペナルティ項を用いても、非零のキネティックオーダー数が 2 個以上の S-system パラメータを見つけることはできる。

3.3.2 動的関数

ペナルティ項が付加されている目的関数の問題を解く際には、しばしば本来の目的関数の項とペナルティ項にそれぞれ動的に変化する係数を乗じることで、効率的に最適化を行う方法が用いられ、本研究でもそのように動的に変化する係数を導入する。ネットワーク推定開始時刻を 0 とし終了時刻を t_{end} 、現在の時刻を t とすると、PUNN モデルの学習時に用いる目的関数 $E_{P,i}$ は式 (21) のようになる。なお、以下 PUNN モデルの学習時に用いる目的関数として式 (21) を用いることとする。

$$E_{P,i} = c_P(t) \times E'_{P,i} + (1 - c_P(t)) \times P_{P,i} \quad (21)$$

$$c_P(t) = \left(1 - \left(1 - \frac{t}{t_{end}}\right)^2\right) \times 0.4 + 0.5 \quad (22)$$

4. 最適化手法

PUNN モデルを用いた推定手法では遺伝子ネットワークを PUNN で表現し PUNN を学習することで、遺伝子間の制御関係である S-system パラメータを求める。そのため PUNN

の学習精度は PUNN モデルを用いた推定手法の推定精度に大きな影響を与える。文献 11)、12) によると、NN の学習によく用いられる傾き降下法 (Gradient Descent: GD) は PUNN の学習には不向きであるといわれている。そのため遺伝的アルゴリズム (GA) や PSO (Particle Swarm Optimization: 粒子群最適化) などの進化的計算手法を用いて PUNN の学習を行う研究がなされている^{11),12)}。そこで本研究では学習時間と学習精度においてともに良い結果が得られている PSO を用いて PUNN モデルの推定を行う。

4.1 Particle Swarm Optimization: PSO

Kennedy らによって 1995 年に提案された PSO は鳥の群れや魚の群泳における個体は自身の情報だけでなく、群れの情報も用いて行動していることに注目し、各個体が現在の移動方向と群れの中の最良位置への方向を考慮して、次の移動方向を決定することにより、群れをなしながら集団としてより良い位置を探すように振る舞う点を模倣した最適化手法である¹³⁾。PSO において個体は粒子 (particle) と呼ばれ、与えられた問題の解候補を表す問題空間上の座標 (位置) と移動方向を表す速度を持っており、群れをなして D 次元の問題空間を移動することで最適化が行われる。そのため PSO は各粒子の速度を制御し、粒子の位置を移動させることで、与えられた問題の最適解を見つけようとする手法ともいえる。基本的な速度 $\mathbf{v}_i^k = (v_{i,1}^k, \dots, v_{i,D}^k)$ および位置 $\mathbf{x}_i^k = (x_{i,1}^k, \dots, x_{i,D}^k)$ の更新式は次のように定義される。

$$v_{i,d}^{k+1} = wv_{i,d}^k + c_1 \text{rand}_1 \times (p_{i,d} - x_{i,d}^k) + c_2 \text{rand}_2 \times (p_{g,d} - x_{i,d}^k) \quad (23)$$

$$x_{i,d}^{k+1} = x_{i,d}^k + v_{i,d}^{k+1} \quad (24)$$

\mathbf{x}_i^k は粒子 i の k 回移動後の位置、 \mathbf{p}_i は粒子 i の過去の最適化において最良の目的関数値を与える位置 (ローカルベスト)、 \mathbf{p}_g は $\mathbf{p}_i, i = 1, \dots, N_p$ の中の最良の位置、すなわち、群れ全体の過去の最適化における最良の位置 (グローバルベスト) である。 N_p は群れの中の粒子数を表しており、 c_1, c_2 は、それぞれ $\mathbf{p}_i, \mathbf{p}_g$ が次の速度 \mathbf{v}_i^{k+1} に及ぼす影響を示す係数である。なお $w = 0.730, c_1 = c_2 = 1.496$ が良い値であるといわれている¹⁴⁾。 $\text{rand}_1, \text{rand}_2$ は 0 から 1 の一様性乱数を表し、毎回更新時に各粒子、各次元ごとに独立に乱数を発生させる。また速度には上限速度 $v_{max,d}$ と下限速度 $-v_{max,d}$ が定められており、速度がその範囲内に収まるように、速度を次のように修正する。

$$v_{i,d}^{k+1} = \begin{cases} v_{max,d} & \text{if } v_{i,d}^{k+1} \geq v_{max,d} \\ -v_{max,d} & \text{if } v_{i,d}^{k+1} \leq -v_{max,d} \\ v_{i,d}^{k+1} & \text{else} \end{cases} \quad (25)$$

上限速度 $v_{max,d}$ は各粒子の速度の発散を防ぐ働きだけでなく、粒子の高速移動にともなう最良解の見落としを防ぐ働きがある。粒子の移動速度が大きいほど、単位時間に移動する距離が長くなる。連続時間上で考えると、その間の移動経路は過去の最適化として扱われ、 p_i もしくは p_g を更新する可能性がある。しかしながら PSO は離散時間上で群れを制御するため、単位時間中の移動経路のうち始点と終点（現在の時刻と次の時刻の位置）しか過去の最適化として扱えない。そのため始点と終点を結ぶ直線上に最適解があったとしても発見できない。このような見落としの危険性は移動速度が大きいほど高くなり、上限速度を設けることでこの危険性を一定以下に保っている。しかしながら上限速度を小さく設定すると粒子が移動せず最適化が進まないため、見落としの危険性と最適化の進行のトレードオフの関係がある。多くの基礎研究では上限速度 $v_{max,d}$ は値域の最大値に設定されることが多い。これは基礎研究で扱う問題の値域が 0 を中心として対称であるためであり、値域にそのような特徴がない場合には注意が必要となる。そのため本研究では d 次元の値域を $[R_{min,d}, R_{max,d}]$ とすると、 $v_{max,d} = (R_{max,d} - R_{min,d})/2.0$ とする。

この章で説明した PSO は基本的な PSO であり、速度の更新式などに様々な改良を加え、より高精度の解候補を求めることができる手法が提案されている。しかしながら事前実験により、短時間で PUNN の学習を行う際には、最適化の早い段階である程度の解候補を見つけることができる基本的な PSO で十分であると判断したため、本研究では基本的な PSO を用いることとした。

4.2 初期化

一般的に PSO や GA などの多点探索手法では、解候補の初期値として乱数値を用いる。しかしながら、遺伝子ネットワークの結合は疎であるといわれており、キネティックオーダに関するペナルティ項を目的関数に加えることで、その特徴を最適化に導入している。そこで本研究では、その特徴を解候補の初期化時にも導入することで、より効率的な初期化を行う。具体的には初期化するキネティックオーダ数 N_{Init} を $2 \sim N$ の範囲で乱数により決定する。2 個以上に設定した理由は 3.3.1 項で述べた理由と同じである。上限値に関しては、キネティックオーダ数 $2N$ の半分の N を採用した。図 2 に初期化のアルゴリズムを示す。このアルゴリズムは解候補を初期化することに実行され、解候補によって初期化されるキネ

Initialize(*Indiv*)

- (1) 初期化するキネティックオーダ数を N_{Init} に設定する。
- (2) N_{Init} 個のキネティックオーダをランダムに選択し X_a とする
- (3) for すべてのキネティックオーダ i に対して do
- (4) if $i \in X_a$ do
- (5) キネティックオーダ i の初期化
- (6) end if
- (7) end for

図 2 初期化のアルゴリズム

Fig. 2 The initialize algorithm.

ティックオーダ数が異なる。

4.3 ローカルサーチ

多くの最適化問題において、その問題に特化したヒューリスティクスやローカルサーチを導入することによって、より良い解を求められることが多くある。遺伝子ネットワーク推定問題においても、例外ではなく遺伝子ネットワークが疎な関係であるというヒューリスティクスを用いたローカルサーチ Hill Climbing Local Search (HCLS) が提案されている⁷⁾。HCLS では絶対値の小さな順にキネティックオーダを 0 に設定し、評価値が良くなる場合には、解候補を改善する。本研究ではこれに加え、与えられた解候補に $\pm 5\%$ の誤差を付加した新たな解候補を n_{L1} 個生成し、与えられた解候補より評価値が更新された場合のみ、解候補を修正する。これによりキネティックオーダのみを対象としたローカルサーチだけでなく、すべてのパラメータを対象としてローカルサーチを行うことができる。図 3 にローカルサーチのアルゴリズムを示す。このローカルサーチは 2 段階に分かれており、1 段階目はパラメータ全体を対象とするローカルサーチであり、2 段階目は HCLS に対応している。

5. 評価実験の設定

5.1 遺伝子発現量の求め方

PUNN モデルを用いた推定手法の性能を評価するために、従来研究においてよく用いられている 5 つの遺伝子からなる遺伝子ネットワークを用いて評価実験を行う。文献 6) と同様に実験に用いる遺伝子発現量の時系列データを各遺伝子の初期遺伝子発現量と目標 S-system パラメータ（ともに文献 6) と同じ値）を用いて、微分方程式 (2) を解くことによって得た。なお与えられる時系列データの量が十分でないと、モデルの自由度が高くなり解候補は増大する。そのため適切な解を見つけ出すためには十分な量の遺伝子発現量の時系列データが必要であると考え、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準備した。また

```

LocalSearch(Indiv)
  <第 1 段階>
  (1) Indiv を Indiv0 にコピーする .
  (2) Loop  $i = 1, \dots, N_{L1}$  do
  (3)   for すべての変数  $j$  に対して do
  (4)      $0.95 \sim 1.05$  の乱数値を生成し  $R_1$  とする .
  (5)      $Indiv_{i,j} = Indiv_{0,j} * R_1$ 
  (6)   end for
  (7) end Loop
  (8)  $Indiv_i$  ( $i = 0, \dots, N_{L1}$ ) の中で評価値が一番良い解候補を Indiv に設定する
  <第 2 段階>
  (9) Indiv のキネティックオーダを絶対値の昇順にソートして,  $K(i)$  ( $i = 1, \dots, 2N$ ) とする .
  (10) Loop  $i = 1, \dots, 2N$  do
  (11) Indiv の  $K(i)$  を 0 にセットした新しい解候補 Indiv'
      を生成する
  (12) もし,  $f(Indiv') < f(indiv)$  ならば, Indiv' を Indiv
      とする .
  (13) end Loop

```

図 3 ローカルサーチのアルゴリズム
Fig. 3 The local search algorithm.

時系列データは遺伝子ごとに 11 サンプルポイントになるように適当に間引き, これを 15 セット用意するので, 各遺伝子上の時系列データは 165 サンプルポイントとなる .

5.2 発現変化量の求め方

NN モデルを用いた推定手法が提案された研究⁴⁾では, 遺伝子発現量からスプライン補間を用いて発現変化量を求めていた . しかしながら PUNN モデルを用いた推定手法では求めた発現変化量を教師信号として学習を行うことで遺伝子ネットワーク推定を行うため, 求められる発現変化量に含まれる誤差と PUNN モデルを用いた推定手法の推定能力に何からの関係があると考えられる . そこで本研究では以下の方法で求めた精度の異なる 3 つの発現変化量を用いて評価実験を行う . なお遺伝子発現量データと同様に, 発現変化量の時系列データとして 165 サンプルポイント (= 11×15) のデータを求める

(1) 補間した時系列データの発現変化量

与えられた遺伝子発現データを補間して発現変化量を求める方法であり, S-system モデルを用いた推定手法に与えられる発現量データと同じデータを扱う . またこの方法で求めた発現変化量を用いる PUNN モデルを PUNN モデル (補間) と呼ぶ .

(2) 正確な時系列データを用いた差分量

与えられた遺伝子発現量から差分法を用いて発現変化量を求める方法であり, より細かい

遺伝子発現量データが生物学的実験により得られると仮定する場合である . 評価実験の際には遺伝子発現量データを 11 ポイントに間引く前の発現量データを用いて発現変化量を求めた . なお, これは目標とする遺伝子ネットワークの情報を S-system モデルを用いた推定手法より多く与えていることになる . この方法で求めた発現変化量を用いる PUNN モデルを, PUNN モデル (差分量) と呼ぶ .

(3) 正確な発現変化量

与えられた発現変化量を用いる方法であり, 生物学的実験により遺伝子の発現変化量が得られると仮定する場合である . 評価実験の際には, 正確な遺伝子発現量の時系列データと目標 S-system パラメータを用いて発現変化量を求めた . なお正確な発現変化量とするため実験誤差は含まれないものとする . これは目標遺伝子ネットワークの情報のうち発現量データに加え, 発現変化量まで用いて遺伝子ネットワーク推定を行っており, 上記の 2 つよりもさらに多くの情報を PUNN モデルを用いた推定手法に与えているといえる . この方法で求めた発現変化量を用いる PUNN モデルを, PUNN モデル (正確) と呼ぶ . なお参考までに, この方法で求めた発現変化量と比べた場合, 補間した時系列データの発現変化量の絶対誤差は 3.375, 正確な時系列データを用いた差分量の絶対誤差は 3.796×10^{-5} であった .

5.3 比較対象と比較方法

NN モデルを用いた推定手法は遺伝子間の制御関係として, 関係性の有無および正負の 3 種類に分けて推定を行うが, PUNN モデルを用いた推定手法は制御関係として S-system パラメータを求めており, 2 つの推定手法で出力結果が異なる . そのため比較対象として, NN モデルを用いた推定手法は用いず S-system モデルを用いた推定手法のみとした . S-system モデルを用いた推定手法で用いる最適化手法として, Christian らの研究¹⁵⁾では実数値遺伝子アルゴリズム (real-valued GA) や進化的戦略 (standard evolutionary strategy), PSO などの複数の進化的計算手法が用いられている . その研究ではキネティックオーダに関するペナルティや AIC は導入されていないが, CMA を用いた進化的戦略, UNDX を用いた実数値遺伝的アルゴリズム, Differential Evolution の 3 つの最適化手法で良い結果を得ている . このことから S-system モデルを用いた推定手法では, それら 3 つの手法のいずれかを用いるほうが良い推定精度を得られる可能性がある . しかしながら, 本研究では PUNN モデルを用いた遺伝子ネットワーク推定手法の提案を目的としており, 同じ最適化手法を用いて評価実験を行う必要がある . そのため, 本研究では PUNN の学習に適用されている PSO を共通の最適化手法とした .

また S-system モデルを用いた推定手法と PUNN モデルを用いた推定手法ではパラメータ

の目的関数が異なるため、両者の目的関数値を比較することに意味はない。そのため実験結果および考察の際には、目的関数値ではなく式 (26) のパラメータ誤差 $PE(\mathbf{P})$ を用いる。ただし、 $\mathbf{P} = \{\alpha_i, \beta_i, g_{i,j}, h_{i,j}\} (i, j = 1, \dots, n)$ であり、 $\alpha_i^t, \beta_i^t, g_{i,j}^t, h_{i,j}^t$ は目標 S-system パラメータを示す。なお、 $\sum (|\alpha_i^t| + |\beta_i^t|) + \sum \sum (|g_{i,j}^t| + |h_{i,j}^t|) = 114$ であった。

$$PE(\mathbf{P}) = \sum_{i=1}^N (|\alpha_i - \alpha_i^t| + |\beta_i - \beta_i^t|) + \sum_{i=1}^N \sum_{j=1}^N (|g_{i,j} - g_{i,j}^t| + |h_{i,j} - h_{i,j}^t|) \quad (26)$$

6. 評価実験

6.1 評価実験 1：評価精度の検証

S-system モデルの遺伝子ネットワーク推定において、目標 S-system パラメータを求めることが目的となる。しかしながらパラメータ誤差を計算できないために、目的関数を用いてパラメータの良さを評価している。そのため、ある 2 つのパラメータが与えられたとき、それらのパラメータ誤差の大小関係と目的関数値の大小関係は同じであることが望まれる。

そこで各モデルにおいて、パラメータ誤差と目的関数値の関係を図示し、どの程度それらの大小関係が一致しているかを調べる。各モデルにおけるパラメータ誤差と目的関数値の関係を図 4 に示す。なお本評価実験に限り、サブ問題に分割することによって S-system モデルの評価精度がどのように変化するか調べるために、S-system モデルをサブ問題に分割するモデル (S-system モデル (分割) と呼ぶ) とサブ問題に分割しないモデル (S-system モデル (非分割) と呼ぶ) の 2 つに分けてパラメータ誤差と目的関数値の関係を示した。図 4 (a) と図 4 (e) は、パラメータ誤差に対してほぼ単調増加傾向が見られる。しかしながらそれ以外では、パラメータ誤差が一定値以下になると目的関数値が一定になっていることが分かる。

6.2 評価実験 2：有効性の検証

ある問題をモデル化するうえで、解候補を正確に評価することは重要であるが、正確性を求めたために、問題空間の複雑化や評価時間の増加などの原因により最適化において十分な解を得ることができなければ、その問題を解決したとはいえない。そこで 4 章で述べた最適化手法を用いて、実際に遺伝子ネットワーク推定問題を解くことで、PUNN モデルを用いた推定手法の有用性を検証する。実験条件は文献 6) と同様に以下のように設定した。S-system パラメータの探索領域は、 α_i と β_i が 0.0 から 20.0、 $g_{i,j}$ 、 $h_{i,j}$ は -3.0 から 3.0 とした。また計算コストを削減するため、 $g_{i,j}$ 、 $h_{i,j}$ が閾値 (1.0×10^{-3}) 以下の場合には 0 を返すようにした。粒子数はともに 60 個体とし、ローカルサーチの際に用いる n_{L1} は

20 とした。実験環境は AMD Athlon(tm) 64 × 2 Dual Core Processor 3800+ 2.01 GHz, 1.00 GB のコンピュータを用いた。30 回試行を行った結果を表 1 に示す。実行時間は 1 回の試行に要した時間であり、1 つのサブ問題ではなく、すべての S-system パラメータを推定するために要した時間を示す。評価回数とはパラメータを評価した回数であり、目的関数の計算回数を示す。誤差はパラメータ誤差を示しており、30 回試行中の平均値、標準偏差、最良値を示している。なお本評価実験では実行時間の上限値を 5 時間 (各サブ問題を 1 時間以内で解く) としており、移動回数 500 回以上において、S-system モデルを用いた推定手法は時間内に解くことができなかった。

6.3 評価実験 3：初期化とローカルサーチの影響の検証

本研究では PUNN モデルを用いた推定手法を提案するとともに、初期化とローカルサーチも導入している。そこで初期化やローカルサーチを導入することで、最適化がどのように変化するか確認するために、初期化とローカルサーチを導入せずに遺伝子ネットワーク推定問題を解くことで検証を行う。実験設定はすべて 6.2 節の実験と同じである。実験結果を表 2 に示す。なお初期化とローカルサーチを導入した実験結果は表 1 に示してある。

7. 考察

7.1 考察 1：評価精度の検証

任意の 2 つのパラメータにおいて、パラメータ誤差の大小関係と目的関数値の大小関係が同じとき、パラメータ誤差と目的関数値は単調増加になることから、S-system モデル (非分割) (図 4 (a)) と PUNN モデル (正確) (図 4 (e)) はつねにパラメータ誤差と目的関数値の大小関係が同じであることが分かる*1。

一方それ以外のモデルでは、単調増加になっている領域もあるが、パラメータ誤差が一定値以下になると、目的関数値が一定値になっており、パラメータ誤差と目的関数値の大小関係がつねに一致しているとはいえない。目的関数値が一定値の領域では、パラメータを評価することができないため、一定値になるときのパラメータ誤差の値が、そのモデルにおける評価可能なパラメータ誤差の最小値となる。そのため本研究ではその値を各モデルにおける評価の精度を示す指標とし、最小評価可能ポイントと呼ぶ。なお S-system モデル (非分割) や PUNN モデル (正確) のように目的関数値が一定値になる領域が存在しない場合は、

*1 なお図中には示していないが、パラメータ誤差を 10^{-14} まで設定し実験を行い、つねに単調増加であることを確認している。

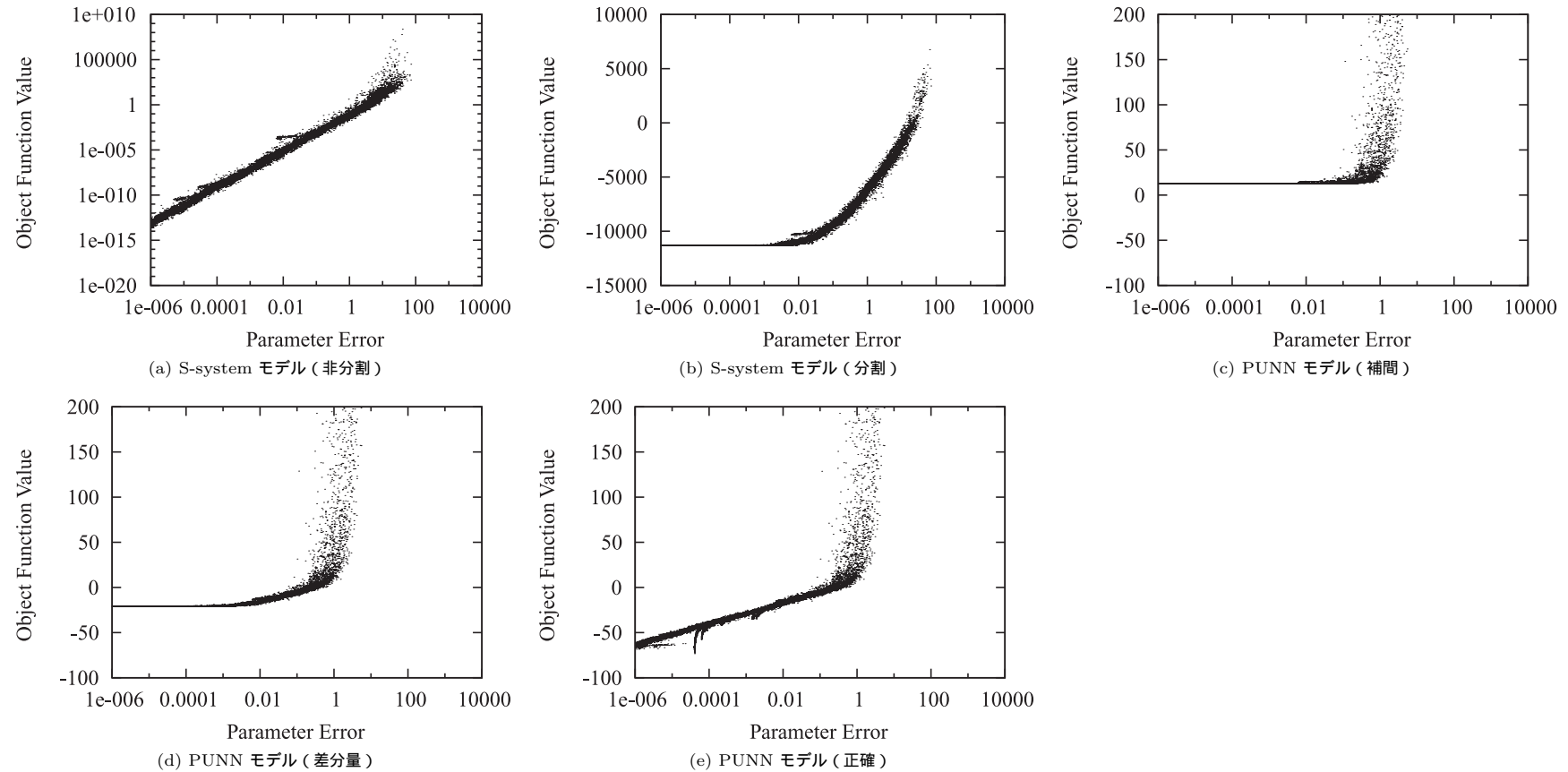


図 4 各モデルにおける目的関数値とパラメータ誤差の関係
 Fig. 4 The relations between objective function value and parameter's error in each model.

最小評価可能ポイントが 0 に存在するものとする。図 4(b) から (d) より、S-system モデル (分割), PUNN モデル (補間), PUNN モデル (差分) の最小評価可能ポイントは、それぞれ約 0.01, 1, 0.01 であることが読み取れ、PUNN モデル (補間) は S-system モデル (分割) よりも評価精度が悪いことが分かる。また PUNN モデル (補間), PUNN モデル (差分), PUNN モデル (正確) と評価精度が良くなっており、使用する発現変化量に

含まれる誤差を少なくすることにより、評価精度を良くできることが分かる。

次に S-system モデル (分割), PUNN モデル (補間), PUNN モデル (差分) はなぜ最小評価可能ポイントが 0 でないかについて考察する。S-system モデルにおいて時刻 t における遺伝子 i の発現量 $X_{i,t}$ は、 $X_{j,t+1}$ ($j = 1, \dots, N$) に影響を与える。そのため $X_{i,t-2}$ が $X_{i,t}$ に与える影響は、遺伝子 i の発現量として伝わる直接的な影響と他の遺伝子の発現量

表 1 各推定手法の遺伝子ネットワーク推定結果

Table 1 The results of inference of genetic networks in each method.

移動回数	使用モデル	実行時間 [s]	評価回数	誤差 (平均)	誤差 (標準偏差)	誤差 (最小)
500	S-system モデル	19891.73	1518420	6.2197	14.7249	0.0571
	PUNN モデル (補間)	123.43	1251817	3.2900	1.6494	1.1332
	PUNN モデル (差分)	132.08	1249353	2.3724	2.0382	0.2476
	PUNN モデル (正確)	123.38	1253127	2.1277	1.6807	0.2361
1000	S-system モデル	-	-	-	-	-
	PUNN モデル (補間)	181.15	2092374	2.6717	1.5272	1.1284
	PUNN モデル (差分)	181.66	1947153	1.5281	1.8697	0.0701
	PUNN モデル (正確)	173.98	2035241	1.5609	1.8026	0.0624
5000	S-system モデル	-	-	-	-	-
	PUNN モデル (補間)	667.95	9034026	1.8170	0.8323	1.1867
	PUNN モデル (差分)	571.29	7398901	0.3022	0.8187	0.0118
	PUNN モデル (正確)	562.43	6890762	0.6055	1.3121	0.0000
10000	S-system モデル	-	-	-	-	-
	PUNN モデル (補間)	1317.66	18064577	1.6636	0.4853	1.1972
	PUNN モデル (差分)	1102.35	14045943	0.2909	1.2666	0.0118
	PUNN モデル (正確)	1017.19	12804644	0.2889	0.8469	0.0000
50000	S-system モデル	-	-	-	-	-
	PUNN モデル (補間)	6301.80	90134959	1.5278	0.1054	1.2124
	PUNN モデル (差分)	4727.22	60929488	0.0289	0.0317	0.0118
	PUNN モデル (正確)	4728.00	59369760	0.0190	0.0401	0.0000

表 2 初期化とローカルサーチの影響

Table 2 The influence of initialization and local search.

移動回数	使用モデル	実行時間 [s]	評価回数	誤差 (平均)	誤差 (標準偏差)	誤差 (最小)
500	PUNN モデル (補間)	78.18	342433	11.5304	8.3855	1.1476
	PUNN モデル (差分)	76.22	336350	10.1411	9.1057	1.1739
	PUNN モデル (正確)	63.39	335393	11.0851	9.1508	0.9172
1000	PUNN モデル (補間)	110.65	631494	9.4369	7.3789	0.6392
	PUNN モデル (差分)	100.23	608675	7.2497	7.5438	0.5750
	PUNN モデル (正確)	105.70	605893	7.4485	7.8740	0.1888
5000	PUNN モデル (補間)	320.56	2901637	5.0740	4.4536	1.0741
	PUNN モデル (差分)	315.33	2582085	3.1698	4.8268	0.0118
	PUNN モデル (正確)	306.82	2629450	3.6539	5.0255	0.0004
10000	PUNN モデル (補間)	622.62	5796445	4.1670	4.3715	1.0742
	PUNN モデル (差分)	565.74	5013003	2.8069	5.3814	0.0118
	PUNN モデル (正確)	574.82	4991860	3.9578	6.7442	0.0000
50000	PUNN モデル (補間)	2848.91	27479937	3.2970	4.8458	1.0426
	PUNN モデル (差分)	2560.81	22967389	0.7456	3.4855	0.0118
	PUNN モデル (正確)	2627.30	23272230	1.2351	2.8347	0.0000

を介して伝わる間接的な影響^{*1}の2種類あることが分かる。しかしながら S-system モデル

(分割)では、 $\hat{X}_j(j \neq i)$ を用いることから、他の遺伝子の発現量は X_i と無関係になっている。つまり S-system パラメータによって変化する $X_{i,t}$ の影響のうち、他の遺伝子の発現量を介して伝わる間接的な影響を考慮できていないと考えられる。そのため S-system モデル (分割)では、S-system パラメータを正確に評価できないと考えられる。なお S-system モ

*1 たとえば $X_{1,1}$ の影響が $X_{2,2}$ に伝わり、 $X_{2,2}$ の影響が $X_{1,3}$ へと伝わった場合、 $X_{1,1}$ の影響が $X_{2,2}$ を介して $X_{1,3}$ へ伝わったと見ることができる。

デル(分割)では,補間を用いて他の遺伝子の発現量を仮定していることから,補間が原因であると考えられる.しかしながら補間を用いずに正確な遺伝子発現量を用いて実験したところ補間を用いた場合とまったく同じ結果になったため,最小評価可能ポイントには関係ないと考えられる.PUNN モデル(補間)と PUNN モデル(差分量)においては,使用する発現変化量に含まれる誤差が原因だと考えられる.PUNN モデルにおいて使用する発現変化量に誤差が含まれる場合,その誤差を含めた発現変化量を算出する S-system パラメータを目標パラメータとして,与えられたパラメータの評価を行う.しかしながら誤差の含まれる発現変化量を算出する S-system パラメータと,本来目標とする目標 S-system パラメータとは異なると考えられ,含まれる誤差が大きいほどその差異は大きくなると考えられる.そのため最小評価可能ポイントが 0 とならず, PUNN モデル(補間)のほうが PUNN モデル(差分量)より最小評価可能ポイントが大きいと考えられる.

7.2 考察 2: 有効性の検証

移動回数 500 回において S-system モデルを用いた推定手法と PUNN モデルを用いた推定手法の実行時間を比べると, S-system モデルを用いた推定手法では約 20000 秒であり, PUNN モデルを用いた推定手法は約 120 秒となっており,約 160 倍速いことが分かる.これは PUNN モデルを用いた推定手法ではパラメータの評価の際に微分方程式を解く必要性がないためだと考えられる.

パラメータ誤差について見てみると,平均値,標準偏差ともに PUNN モデルを用いた推定手法のほうが小さくなっており, PUNN モデルを用いた推定手法のほうが平均的に良いパラメータを求めていることが分かる.しかしながら最小値で比較すると, PUNN モデル(補間)を用いた推定手法よりも S-system モデルを用いた推定手法のほうが良いパラメータを求めていることが分かる.しかしながら表 3 に示した従来研究で求められたパラメータと比較すると, PUNN モデル(補間)を用いた推定手法で求められたパラメータでも十分良いパラメータであるといえる.

PUNN モデルを用いた推定手法において,使用する発現変化量に含まれる誤差と求まるパラメータの関係について見てみると,発現変化量に含まれる誤差が少なくなるほど,良いパラメータが求められており,特に移動回数が 5000 回以上の PUNN モデル(正確)を用いた推定手法の場合では,誤差なく目標パラメータを求めることに成功している.PUNN モデルを用いた推定手法では与えられた発現変化量を算出するようなパラメータを求めている.そのため誤差が含まれる発現変化量が与えられると,誤差を含む発現変化量を算出するパラメータを求めることになり,そのようなパラメータは本来目標とするパラメータとは

表 3 従来研究において推定された遺伝子ネットワーク

Table 3 The estimated genetic networks in the previous works.

著者名(参考)	実行環境	実行時間 [s]	誤差
Kimura ⁶⁾	Pentium 3, 1 GHz	17,640	2.463
松村 ¹⁶⁾	Pentium 4, 2.8 GHz	—	9.732
Noman ⁷⁾	Pentium, 1.7 GHz	18,000	0.724

各文献に記載されている最も良いパラメータの誤差を示す.

異なると考えられる.そのため本来目標とするパラメータとは異なるパラメータを目標として最適化を行うため,発現変化量に含まれる誤差が多いほど求まるパラメータの誤差も大きくなると考えられる.

PUNN モデルを用いた推定手法における移動回数の影響について考察すると,どの PUNN モデルを用いた推定手法においても,移動回数が 5000 回以降は,最小パラメータ誤差には大きな変化は見られない.これらの値は各モデルの最小評価可能ポイントに達していることから, PSO を用いることで各モデルにおける限界値まで求められることが分かる.しかしながら,一部例外はあるが,パラメータ誤差の平均値と標準偏差は移動回数が大きくなるにつれて小さくなっていることから,移動回数が少なくても良いパラメータを求めることは可能であるが,移動回数が多いほうが平均的には良いパラメータを求められることが分かる.

ここで S-system モデルを用いた推定手法について,本評価実験の結果と表 3 に示した従来研究の結果を比べると,本評価実験のほうが良いパラメータを求めていることが分かる.本研究では Noman ら⁷⁾のモデルを S-system モデルとして用いているが,最適化手法として Noman らは Trigonometric Differential Evolution (TDE) を用いており,我々は PSO を用いている点が異なる.そのため,用いた最適化手法の違いにより,同じモデルを用いたとしても異なる結果になったと考えられる.そこで, TDE と PSO がどの程度異なるかを,関数最適化問題を用いて調べる.実験に用いた関数は Sphere 関数, Rosenbrock 関数, Rastrigin 関数, Griewank 関数の 4 つである.両手法とも個体数を 60, 移動回数 6000 回として, 30 回試行したときの平均値を表 4 に示す. 5000 次元の Rastrigin 関数を除き,すべての問題において PSO のほうが良い解を求めている.このように最適化手法としての違いが,同じモデルを用いた推定手法にもかかわらず,求められるパラメータに違いが発生したと考えられる.

7.3 考察 3: 初期化とローカルサーチの影響の検証

表 1 に示した初期化とローカルサーチを導入した実験結果と表 2 に示した初期化とロー

表 4 PSO と TDE の性能評価実験結果
Table 4 The results of PSO and TED.

Function	method	dimension					
		12	60	100	500	1000	5000
Sphere	PSO	0.00	0.00	0.00	64434.25	499034.71	8172252.73
	TDE	19557.83	235103.44	448671.00	2691944.07	5555671.49	28289696.84
Rosenbrock	PSO	0.16	47.44	105.14	2794.98	14009.80	280926.55
	TDE	502.64	4488.93	8451.62	46570.39	95648.39	498270.37
Rastrigin	PSO	7.19	296.99	648.86	6434.21	15730.65	121673.75
	TDE	155.36	1259.46	2207.98	11747.66	23833.12	121383.89
Griewank	PSO	0.08	0.06	0.44	609.72	4633.47	73538.40
	TDE	168.24	2100.28	4109.65	24552.81	50107.36	254628.33

カルサーチを導入していない実験結果を比べると、初期化やローカルサーチを導入したほうが、実行時間が長くなっていることが分かる。これはローカルサーチを行う際に、解候補の評価を何度も行うためだと考えられる。最小パラメータ誤差については、導入前後で大きな変化は見られないが、平均値、標準偏差ともに導入後のほうが良くなっている。これはすべての粒子が初期化やローカルサーチにより、少なくとも局所解へと最適化を行えたためだと考えられる。

8. おわりに

本研究では微分方程式モデルの遺伝子ネットワーク推定問題において、Product Unit based NN (PUNN) を用いることで微分方程式を解くことなく、S-system モデルの遺伝子ネットワークを推定できる Product Unit based NN モデル (PUNN モデル) を用いた推定手法を提案した。評価実験の結果、S-system モデルを用いた推定手法と比べ得られた最良パラメータは良くならなかったが、平均値、標準偏差ともに PUNN モデルを用いた推定手法のほうが良くなっており、平均的には PUNN モデルを用いた推定手法のほうが良いパラメータを求められたことを示した。実行時間に関しては約 160 倍の速度向上に成功し、微分方程式を解かないことのメリットが大きいことを示した。また PUNN モデルを用いた推定手法において、与える発現変化量に含まれる誤差が少ないほど最良値、平均値ともに良いパラメータを求められ、誤差が含まれない正確な発現変化量が与えられると仮定した場合、誤差 0 のパラメータを求められたことも示した。

今後の予定としては、与えられた遺伝子発現量の時系列データから、より正確な発現変化量の算出方法を検討するとともに、与えられる遺伝子発現量の時系列データに誤差が含まれ

る場合や、より大規模な遺伝子ネットワーク推定問題に PUNN モデルを用いた推定手法を適用し、PUNN モデルを用いた推定手法の有効性を検証したいと考えている。また文献 17) では、PUNN の学習に基本的な PSO ではなく、共進化 PSO が使われており、PUNN モデルの遺伝子ネットワーク推定にも適用することによって、より高精度の遺伝子ネットワーク推定を行える可能性があり、今後適用したいと考えている。

謝辞 本論文に対して貴重なコメントやご指摘をいただいた担当委員と査読者の方についで感謝の意を表すとともに、数理モデル化と問題解決研究会にて有益な議論をしていただいた先生方に感謝いたします。また、金沢大学大学院の佐藤賢二先生には様々なご指摘をいただきました。この場を借りてお礼申し上げます。

参 考 文 献

- 1) Jong, H.D.: Modeling and simulation of genetic regulatory system: A literature review, *Journal of Computational Biology*, Vol.9, No.1, pp.67-103 (2002).
- 2) Mason, J., Lindsay, P.S., Collins, J.J. and Glass, L.: Evolving complex dynamics in electronic models of genetic networks, *Chaos*, Vol.14, No.3, pp.707-715 (2004).
- 3) Savageau, M.A.: Biochemical systems analysis II — The steady state solution for an n-pool system using a power-law approximation, *J. of Theor. Biol.*, Vol.25, pp.370-379 (1969).
- 4) 木村周平, 園田克樹, 山根総一郎, 松村幸輝, 畠山真里子: 遺伝子ネットワークのニューラルネットワークモデル同定法の提案, 情報処理学会研究報告バイオ情報学, Vol.2007, No.21, pp.79-84 (2007).
- 5) Durbin, R. and Rumelhart, D.: Product units: A computationally powerful and biologically plausible extension to backpropagation networks, *Neural Computation*,

Vol.1, No.1, pp.133–142 (1989).

- 6) Kimura, S., Hatakeyama, M. and Konagaya, A.: Inference of S-system Models of Genetic Networks from Noisy Time-series Data, *Chem-Bio Info. J.*, Vol.4, No.1, pp.1–14 (2004).
- 7) Noman, N. and Iba, H.: Inference of Genetic Networks using S-system: Information Criteria for Model Selection, *Proc. GECCO-2006*, pp.263–270 (2006).
- 8) Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K. and Tomita, M.: Dynamic modeling of genetic networks using genetic algorithm and S-system, *Bioinf.*, Vol.19, No.5, pp.643–650 (2006).
- 9) Tominaga, D. and Horton, P.: Inference of Scale-Free Networks from Gene Expression Time Series, *J. Bioinf. and Comp. Biol.*, Vol.4, No.2. pp.503–514 (2006).
- 10) 小杉幸夫, 亀山啓輔: 層間荷重に解を得る誤差逆伝播法の逆問題への適用, *信学論 D-II*, Vol.77, No.3, pp.600–605 (1994).
- 11) Engelbrecht, A.P. and Ismail, A.: Training Product Unit Neural Networks, *Stability and Control: Theory and Applications*, Vol.2, No.1–2, pp.59–74 (1999).
- 12) Ismail, A. and Engelbrecht, A.P.: Global Optimization Algorithms for Training Product Unit Neural Networks, *IEEE Computer society*, Vol.1, pp.132–137 (2000).
- 13) Kennedy, J., Eberhart, R.C. and Shi, Y.: *Swarm intelligence*, Morgan Kaufmann Publishers, San Francisco (2001).
- 14) Eberhart, R. and Shi, Y.: Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, *Proc. Congress on Evolutionary Computation (CEC2000)*, pp.84–88 (2000).
- 15) Christian, S., Rene, W. and Felix, S.: Comparing Evolutionary Algorithms on the Problem of Network Inference, *Proc. GECCO-2006*, pp.279–286 (2006).
- 16) 松村幸輝, 種田 洋, 木村周平: 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, *情報処理学会論文誌*, Vol.46, No.11, pp.2814–2830 (2005).
- 17) van den Bergh, F. and Engelbrecht, A.P.: Training Product Unit Networks using Cooperative Particle Swarm Optimisers, *IEEE Neural Network*, Vol.1, pp.126–131 (2001).

(平成 19 年 11 月 22 日受付)

(平成 20 年 1 月 9 日再受付)

(平成 20 年 2 月 4 日採録)



村田 裕章 (学生会員)

平成 18 年石川工業高等専門学校専攻科修了。同年金沢大学大学院博士前期課程入学, 現在に至る。生命情報学に興味がある。



越野 亮 (正会員)

平成 12 年金沢大学工学部卒業。平成 14 年同大学院博士前期課程修了。同年富士通(株)入社, 平成 15 年石川工業高等専門学校助手, 現在, 同講師。博士(工学)。人工知能や最適化等の研究に従事。



三田村公智 (学生会員)

平成 19 年金沢大学工学部情報システム工学科卒業。同年同大学院博士前期課程入学, 現在に至る。群知能に興味がある。



木村 春彦 (正会員)

昭和 54 年東北大学大学院工学研究科博士課程修了。工学博士。同年, 富士通(株)入社。昭和 59 年金沢大学経済学部助教授。現在, 同大学工学部情報システム工学科教授。最適コード変換やプロダクションシステムの高速度等の研究に従事。