

## 遺伝的プログラミングを用いた ネットワークサービス処理時間のモデル化

林田 龍一<sup>†</sup> 渡部 康平<sup>†</sup> 大崎 博之<sup>††</sup>

<sup>†</sup> 大阪大学 大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

<sup>††</sup> 関西学院大学 理工学部 情報科学科 〒669-1337 兵庫県三田市学園 2-1

E-mail: <sup>†</sup>{r-hayashida,k-watabe}@ist.osaka-u.ac.jp, <sup>††</sup>ohsaki@kwansei.ac.jp

あらまし 本稿では、ブラックボックスモデリングの一種である遺伝的プログラミングにより、ネットワークサービスの処理時間のダイナミクスをどの程度正確にモデル化できるかを実験により調査する。ネットワークサービスは、通常、計算機（サーバおよびクライアント）や、ストレージ、ルータ、リンクなど、複数の要素によって構成されるシステム上で提供される。計算機およびネットワークの仮想化技術の普及により、システムを構成する個々の要素を正確にモデル化し、これらを結合することでシステム全体のモデルを構築すること（ホワイトボックスモデリング）がより困難になりつつある。一方、ハードウェアおよびソフトウェアの処理速度の向上や、ストレージの低価格化により、サーバもしくはネットワークエッジにおいて大量のログ情報が収集されつつある。そこで本稿では、サーバおよびネットワークエッジにおいて収集されている大量のログ情報を用いて、ネットワークサービスの特性がどの程度正確にモデル化できるかを調査する。トラフィック量の割合やログの計測期間、遺伝的プログラミングのパラメータをさまざまに変化させた時のモデルの適合度を分析することで、遺伝的プログラミングを用いたネットワークサービス処理時間のモデル化の実現可能性を検証する。  
キーワード 遺伝的プログラミング、ブラックボックスモデリング、ネットワークサービス、システム同定

## System Identification of Network Service Processing Time with Genetic Programming

Ryuichi HAYASHIDA<sup>†</sup>, Kohei WATABE<sup>†</sup>, and Hiroyuki OHSAKI<sup>††</sup>

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University Yamada-oka 1-5,  
Suita-shi, Osaka 565-0871, Japan

<sup>††</sup> Department of Informatics School of Science and Technology, Kwansei Gakuin University Gakuen  
2-1, Sanda-shi, Hyogo 669-1337, Japan

E-mail: <sup>†</sup>{r-hayashida,k-watabe}@ist.osaka-u.ac.jp, <sup>††</sup>ohsaki@kwansei.ac.jp

**Abstract** In this paper, we investigate experimentally how accurately the dynamics of processing time of network service is modeled with Genetic Programming (GP), one of black box modeling approaches. Generally, network services are provided on a system that consists of several elements like computers (servers and clients), storages, routers, and links. As computers and network virtualization technology become widespread, it gets more difficult to build a model of the entire system by modeling individual elements of the system and combining them (white box modeling). On the other hand, huge amount of network/server logs are collected at the edge of the network or server because of increasing processing speed of hardware and software, and decreasing the cost of storage. In this paper, we investigate how accurately characteristics of the network service is modeled using huge amount of log information collected at the edge of network and server. By analyzing the fitness of the model when traffic rate, the measurement period of logs and parameters of GP are changed, we clarify feasibility of network services processing time modeling with GP.

**Key words** Genetic Programming, Black Box Modeling, System Identification, Network Service

## 1. はじめに

ルータやサーバのモデル化は、待ち行列理論を利用したモデル化を始め、古くから多くの研究が行われてきた [1]～[4]. 特に待ち行列によるモデル化は、 $M/M/1$  によるルータのモデル化や指数分布に従うサービス時間を持つ待ち行列を連結したジャクソンネットワークによるネットワークのモデル化を始め、今日に至るまで広く用いられてきた.

しかし、ネットワークサービスは、通常、計算機やストレージ、ルータ、リンクなど、複数の要素によって構成されており、個々の要素を正確にモデル化し結合することでネットワークサービス全体のモデル化をするようなホワイトボックスモデリングの手法ではモデル化が困難になりつつある. 上述した待ち行列理論によるモデルを利用したモデル化では、ルータの挙動やトラヒックの特性を個別に解析することはできるが、これらの要素が互いにどのように影響し合い、それによってどのような挙動が起こるかまでモデル化することは容易ではない.

このような複数の要素からなり、ホワイトボックスモデリングによるモデル化が困難な対象に用いられるモデル化手法として、システムの詳細な構造や各要素間の依存関係を明らかにせずともモデル化できるブラックボックスモデリングが存在する [5], [6]. ブラックボックスモデリングでは、各構成要素の挙動を詳細にモデル化し、それらを組み合わせることで全体の挙動を明らかにしようとするホワイトボックスモデリングとは対照的に、各構成要素間の詳細な関係を明らかにすることなく、全体を一つのシステムと見なし、入力と出力の関係を直接モデル化する.

ブラックボックスモデリングの一手法であり、システム工学の分野で適用され始めている手法として、遺伝的プログラミング (GP; Genetic Programming) が存在する [7]～[18]. 遺伝的プログラミングとは、生物が自然淘汰により進化していく過程を模倣し、関数やプログラムを木構造によって表して解を探索する進化的アルゴリズムであり、1992 年に Koza によって提案された. 遺伝的プログラミングでは、木構造で表現した関数やプログラムを一つの個体として扱い、多数の個体を交叉や突然変異によって淘汰することでより良い個体を選出する.

一方、サーバやネットワークでは様々なログが生成されており、多くの場合それらは簡単な統計情報の算出にのみ利用されているが、それらはブラックボックスモデリングを行うためのデータとして有効に利用できる可能性がある. ネットワークモニタリングではルータにおけるパケットの流量、パケット廃棄率などの情報を得ることができ、サーバモニタリングでは RTT (Round Trip Time) やアプリケーション・プロトコルの状態に関する情報を得ることができる. 例えば、パケットの流量やプロトコルの状態はパケット遅延と密接な関係にあるため、パケットの流量やプロトコルの状態を入力、パケット遅延を出力として遺伝的プログラミングでそれらの間の関係を学習することで、入出力

間のモデル化を行うことができると期待される.

本稿では、複数の要素からなり、内部構成が部分的に不明なルータやネットワークサービスを遺伝的プログラミングによりブラックボックスモデリングすることで、サーバやネットワークに関するログから複数の要素間の関係を同定する手法の実現可能性を実験によって調査する. 具体的には、遺伝的プログラミングによって、ネットワークサービスにおける要求トラヒック量とネットワークサービス処理時間の関係をモデル化する. ネットワークのシステムパラメータやログの計測期間、遺伝的プログラミングのパラメータをさまざまに変化させた時のモデルの適合値を分析し、遺伝的プログラミングによってどの程度ネットワークサービス処理時間のモデル化が可能であるかを分析する.

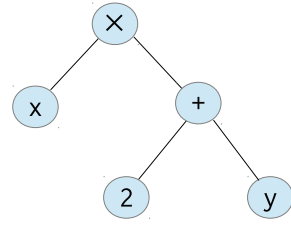
本稿の構成は以下の通りである. まず 2 章では、遺伝的プログラミングについて説明する. さらに 3 章では、遺伝的プログラミングによるネットワークサービス処理時間のモデル化の手法を説明する. 4 章では、 $M/M/1$  待ち行列システムおよび 4 ノードのスタートポロジネットワークを例として、トラヒック量とネットワークサービス処理時間の関係を遺伝的プログラミングによってモデル化し、その時のモデル化精度を分析する. 最後に、5 章で本稿のまとめと今後の課題を述べる.

## 2. 遺伝的プログラミング

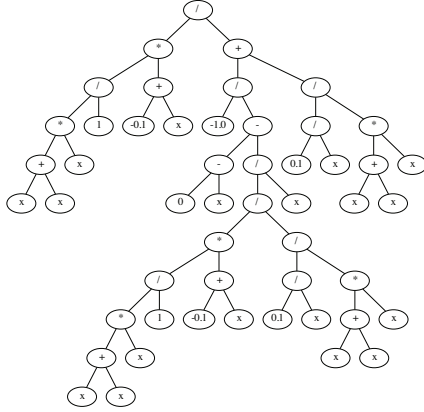
遺伝的プログラミングとは、遺伝的アルゴリズム (GA; Genetic Algorithm) の遺伝子型を式木で表された構造的な表現が扱えるように拡張することで、プログラムの生成や関数同定への利用を可能にした進化的アルゴリズムの一手法である [19]. 遺伝的アルゴリズムと同様に、生物の遺伝子が繁殖や自然淘汰により進化する過程を模倣して、個体の選択と繁殖を繰り返すことで解を探索する. ここでは、遺伝的プログラミングの概要を簡単に紹介する.

遺伝的プログラミングでは、木構造で表現された各個体から交叉と突然変異及び複製によって新しい世代の個体を作り出す処理とその中からよりよい個体を複数選出する処理を交互に繰り返すことで解の探索を行う. 例えば、遺伝的プログラミングを関数同定に利用する場合、木構造を持つ一つの個体が一つの関数を表現する. 例えば、図 1 に示すように木構造の節に四則演算の演算子を、葉に定数または変数を配置することで関数を表現することができる. この場合、探索される解空間は有理関数となる. ランダムに生成された第 0 世代の個体群を元に、交叉処理ではランダムに選択された 2 つの個体の部分木を取り替えることで次の世代の新たな個体を生成し、突然変異ではランダムに選択された個体の部分木をランダムに生成された木に置換することで新たな個体を生成する. 交叉及び突然変異の例を図 2 に示す. また、複製では、個体をそのまま次の世代の個体として引き継ぐ.

個体の優劣は、事前に与えられた入出力に関する教師用データの入出力関係を、各個体がどれだけ表現できている

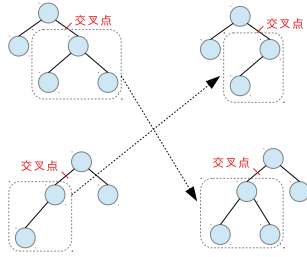


(a)  $f(x, y) = x \cdot (2 + y)$  を表す木構造

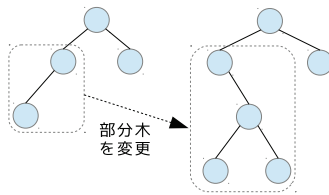


(b) M/M/1 待ち行列システムのブラックボックスモデリングにより得られた有理関数を表す木構造

図 1 有理関数を表す木構造の例



(a) 交叉の例



(b) 突然変異の例

図 2 遺伝的プログラミングにおける遺伝的操作の例

かを意味する適合度を評価することで判断される [20]. 関数同定では、各個体が表現する関数に教師用データの変数値を代入することで得られた値と教師用データの出力値の間の平均自乗誤差を算出して適合値とし、適合値がより小さい個体ほど適合度の高い個体と判断する方法が広く用いられる. 新しい世代の個体を生成するために用いられる個体は、適合度がより高い個体を優先的に選択した一部の個体

のみから生成し、その選択方式にはランキング方式、ルーレット方式、トーナメント方式などがある.

### 3. 遺伝的プログラミングによるネットワークサービス処理時間のモデル化

本稿では、ネットワークサービス処理時間のモデル化を、ネットワークの計測ログやサーバのログから取得できる遅延やトラヒックといった値を入力とする関数として同定する問題として定式化する. ネットワークやサーバで計測できるログの中には、互いに依存関係にあるものが存在し、その関係は関数で表現することができると考えられる. 例えば、ネットワークを流れるトラヒックの量が増加すると輻輳が発生し、ルータにおけるキューイング遅延が増加することによりエンド・エンド間の遅延は増加する. 他にも、サーバの CPU 負荷と処理時間、トラヒック量とパケット廃棄率なども相互に依存している.

ログとして計測された計測値を教師データとして遺伝的プログラミングによるシステム同定を行うことで、ネットワークサービスの挙動をモデル化できると期待できる. 前述のとおり、ネットワークサービスは計算機やストレージ、ルータ、リンクなど複数の要素によって構成されており、トラヒック量と処理時間の依存関係を始め、ネットワークサービスの挙動をホワイトボックスモデリングによって明らかにすることは容易ではない. しかし、ブラックボックスモデリングの一手法である遺伝的プログラミングであれば、サーバやネットワークで生成される様々なログを利用して、その間の依存関係を関数として同定することができる可能性がある. ネットワークにおける計測データ  $i$  ( $i = 0, 1, 2, \dots, n$ ) に関するログのうち、 $k$  ( $k = 0, 1, \dots, m$ ) 番目の計測値を  $x_i^k$  とし、計測データ  $j$  ( $j = 0, 1, 2, \dots$ ) の同時刻の計測値を  $y_j^k$  とする. この時、 $x_i^k$  及び  $y_j^k$  の関係は以下のように記述できる.

$$y_j^k = f_j(x_0^k, x_1^k, \dots, x_n^k)$$

関数  $f_j$  を遺伝的プログラミングによって同定する. 同定によって推定された関数  $\hat{f}_j$  の適合値  $e$  は、教師用データに対する  $y_j^k$  の推定値と  $y_j^k$  の実際の値との平均自乗誤差として定義する.

$$e = \frac{\sum_{k=0}^m (y_j^k - \hat{f}_j(x_0^k, \dots, x_n^k))^2}{m}$$

上式で定義される適合値を元に個体を選別し、関数同定を行うことでネットワークサービスのトラヒック量と処理時間の関係を遺伝的プログラミングによってモデル化することができる.

## 4. 実験

### 4.1 トラヒック量とネットワークサービス処理時間の関係のモデル化

前述の通り、ネットワークサービスのモデル化は遺伝的

表 1 遺伝的プログラミングのパラメータ設定

パラメータ	設定値
1 つ世代に含まれる個体の数	100
次の世代の計算に利用する個体の数	20
個体の選抜	適合値で重み付けした ルーレット方式
部分的な突然変異の生起確率	60[%]
交叉の生起確率	20[%]
複製の生起確率	15[%]
突然変異の生起確率	5[%]
木構造の葉に定数が現れる割合	20[%]
終了条件	20000 世代まで計算

プログラミングを用いたシステム同定の問題として定式化され、様々なログからネットワークサービスのダイナミクスをモデル化することが可能であると期待されるが、本稿では特に、ネットワークサービスにおけるトラフィック量と処理時間の関係に焦点を当て、モデル化を行う。つまり、 $x_i^k$  は、ネットワークの各エッジノードにおける入力トラフィック量及び出力トラフィック量とし、 $y_j^k$  は、パケットやネットワークサービスに対するリクエストが処理され、パケットの転送やリクエストされた処理が完了するまでの時間である。

#### 4.2 M/M/1 待ち行列システムのモデル化

遺伝的プログラミングを用いたネットワークサービスのトラフィック量と処理時間の関係のモデル化がどの程度の精度で実現されるかを分析するために、まず、最も単純な例として、M/M/1 待ち行列システムのトラフィック量と処理時間の関係を遺伝的プログラミングによってモデル化し、その精度を確認する。M/M/1 待ち行列システムのトラフィック量と処理時間の関係は、トラフィック量に相当する入力レートを  $\lambda$  とし、処理時間に相当する平均系内滞在時間  $T$  とすると、両者の関係は退去レート  $\mu$  を使って  $T = 1/(\mu - \lambda)$  と表されることが解析的に知られているが [21]、これらの関係を遺伝的プログラミングによって再現可能であることを確認する。

シミュレーションにより M/M/1 システムの単位時間あたりのパケット到着量及び系内滞在時間を計測し、計測値を教師用データとして遺伝的プログラミングにより到着量と系内滞在時間の間の関数関係を同定する。シミュレーションでは、退去レート  $\mu$  は 1.0 [s/packet] とし、平均到着レートを 0.01 [s/packet] から 0.99 [s/packet] まで 0.01 刻みで変化させ、計測期間 10,000 [s] の間に到着したパケット数から算出した平均到着レートと到着したパケットの平均処理時間を求めた。これらの計測は 10 回繰り返し、計測から得られたログを教師用データとして遺伝的アルゴリズムによりシステム同定を行う。解空間は 2 章で示した例と同様に有理関数とし、木構造の節部分は四則演算の演算子とした。また、遺伝的プログラミングのパラメータ設定を表 1 に示す。

遺伝的プログラミングにより M/M/1 待ち行列システムの入力レートと系内滞在時間の関係をモデル化し、関数  $\hat{f}$  を

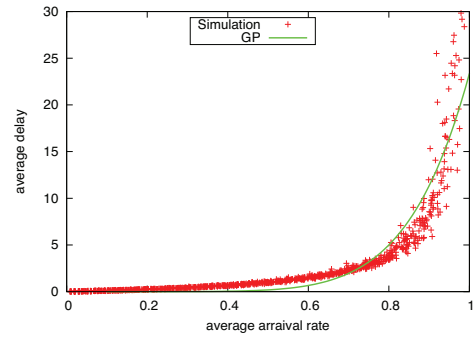


図 3 M/M/1 待ち行列システムにおける系内滞在時間：シミュレーションにおける計測値と遺伝的プログラミングにおける推定値

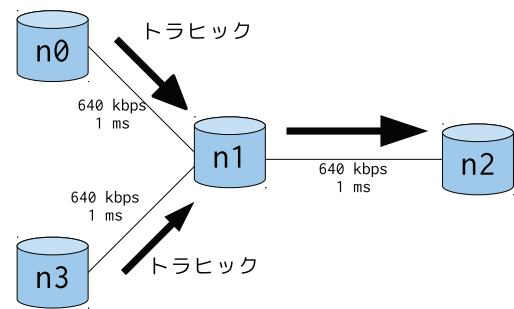


図 4 モデル化対象のネットワーク

同定した。同定した  $\hat{f}$  を表す木構造を図 1(b) に示す。図 3 では関数  $\hat{f}$  と計測により得られた計測値をプロットしているが、図から入力レートと系内滞在時間の関係をうまくモデル化できていることが確認できる。

#### 4.3 4 ノードのスタートボロジネットワークのモデル化

前節で示した M/M/1 待ち行列システムを拡張し、より複雑な 4 ノードのスタートボロジネットワークにおいて実験を行う。モデル化対象のネットワークは、図 4 に示すように  $n_0$ ,  $n_1$ ,  $n_2$ ,  $n_3$  の 4 ノードからなり、ポアソン過程に従い送信されるパケットが  $n_0$  および  $n_3$  から  $n_2$  に向けて送信される。パケット送信レートは、平均 2.0 [s] の指数分布に従う間隔で変化させた。パケット送信レートは  $[R, R + 320]$  [kbps] の一様分布に従い、 $R$  は 0 [kbit/s] から 144 [kbit/s] まで 16 [kbit/s] 刻みで 10 段階に変化させた。また、パケットサイズは 1500 [byte] 各リンクの帯域は 640 [kbit/s]、伝搬遅延は 1 [ms] とした。

ネットワークのエッジノード  $n_0$ ,  $n_2$  および  $n_3$  において、入出力トラフィック量を計測した。あわせて、各パケットが  $n_0$  および  $n_3$  から送出されてから  $n_2$  に到着するまでの時間を計測した。トラフィック量の計測では、1,000 [s] の計測期間中に到着したパケット数を記録した。また、パケット到着までに要する処理時間の計測では、計測期間中に  $n_0$  および  $n_3$  から送出されたすべてのパケットに関して処理時間



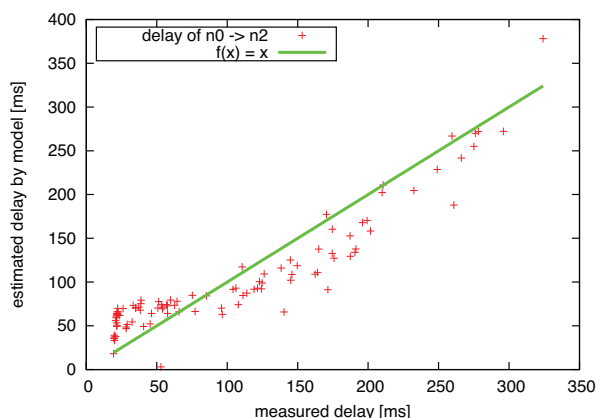


図 5 モデルによる推定値と計測による実測値の関係

を計測し、その平均を記録した。パケット送信レートを決定する  $R$  毎に、それぞれの計測を 10 回ずつ行い、計 100 回の計測を行った。

計測によって得られた各エッジノードにおけるトラヒック量及びパケット到着に要する時間のログを教師用データとして、遺伝的プログラミングを利用してシステム同定を行った。遺伝的プログラミングのパラメータ設定は前節の M/M/1 待ち行列システムのモデル化で用いたパラメータと同じものを用いた。教師用データとは別に上述のシミュレーションと同様のシミュレーションによってログを取得し、同定によって導出された関数にトラヒック量のログの値を代入することで得られた推定の処理時間と実際に計測した処理時間を、それぞれ縦軸と横軸に表示した散布図を図 5 に示す。モデル化精度が高いほど傾き 1 の直線に近づくことになる。図 5 から適当なモデル化ができていることが確認できる。

計測期間がモデル化の精度に与える影響を検証するために、計測期間を 10~10,000 [s] に変化させて、適合値の変化を計測した。計測期間を除くその他のシミュレーション条件は、上述のシミュレーションと同様とした。計測期間と適合値を図 6 に示す。検証に利用したログは教師用データに利用したログとは異なるログであることに注意する。計測期間が 10 [s] の時の適合値を除けば、計測期間が増加するに連れて、平均自乗誤差が小さくなり、モデル化精度が向上していることが確認できる。これは、計測期間が長くなることで、計測期間の前後のトラヒック量の影響が減少し、トラヒック量に対して平均処理時間が一意に決まりやすくなるためであると考えられる。

また、モデル化に用いる遺伝的プログラミングの世代数の影響を調査するために、世代数を 200~200,000 に変化させて、適合値を計測した (図 7)。計測期間は 1,000 [s] とし、その他のパラメータは上述のシミュレーションと同一である。世代数が増加するにつれ適合度が改善していることが確認でき、その改善の度合いは世代数が小さい領域で非常に顕著である。図 7 で縦軸が対数表示になっていることに注意する。これは、遺伝的プログラミングでは解の探索の

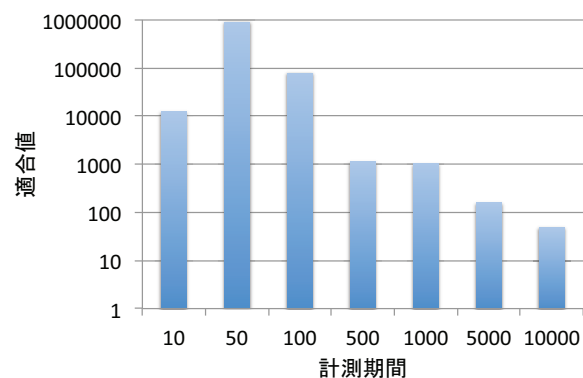


図 6 計測期間を変化させた場合のモデルによる推定値と計測による実測値の間の平均自乗誤差

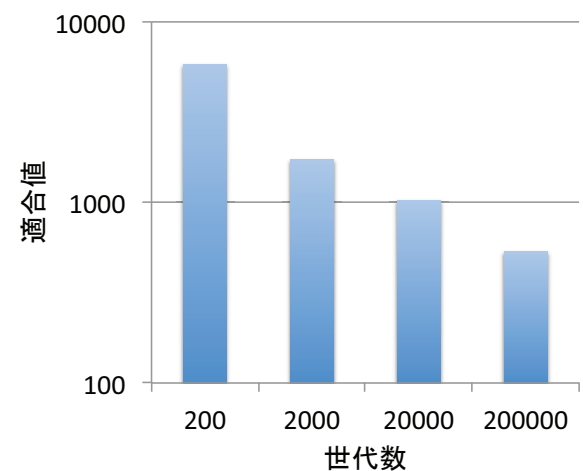


図 7 遺伝的プログラミングの世代数を変化させた場合のモデルによる推定値と計測による実測値の間の平均自乗誤差

初期において適合値を大きく減少させるためであると考えられる。このことは、世代数と最も適合値が低い個体の適合値の関係からも確認できる (図 8)。さらに、世代数の増加に伴うモデル化精度の向上の効果は、遺伝的プログラミングにおいて一世代に含まれる個体の数を増加させることでも同様に得られる。図 9 は、一世代に含まれる個体数を 25~75 に変化させた場合の適合値を示す。一世代に含まれる個体数を 25 から 50 に増加させると、適合値は世代数の増加と同様の効果により一旦急激に減少するものの、さらに個体数を増加させても適合値はほとんど減少しないことが確認できる。

## 5. おわりに

本稿では、サーバやネットワークエッジにおけるログを用いた遺伝的プログラミングによるネットワークサービス処理時間のモデル化の実現可能性を実験によって調査した。ログを用いたネットワークサービスのモデル化を、遺伝的プログラミングによる関数同定問題として定式化し、トラヒック量とネットワークサービス処理時間の関係をモデル化できることを示した。また、遺伝的プログラミングのパ

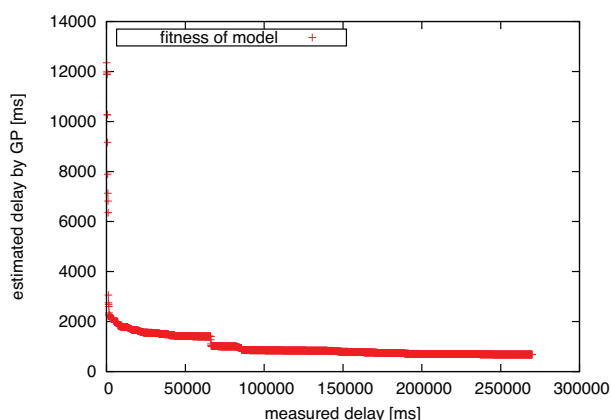


図 8 世代数と最も適合値が低い個体の適合値の推移

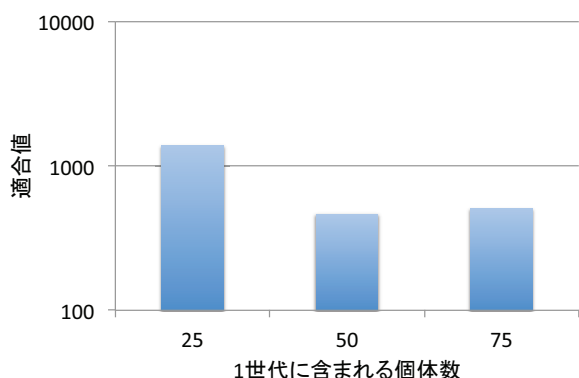


図 9 遺伝的プログラミングで一世代に含まれる個体数を変化させた場合のモデルによる推定値と計測による実測値の間の平均自乗誤差

ラメータを変更して検証を行うことで、遺伝的プログラミングの世代数及び一世代に含まれる個体数がモデル化精度に大きく影響することを示した。計測期間を変化させてモデル化精度を検証した実験から、モデル化したい計測項目の関係が一意でない場合にモデル化精度が低下することがわかったため、今後の課題として、各計測値の前後の計測値についても遺伝的プログラミングの教師用データとして与え、ネットワークサービスのダイナミクスをより詳細にモデル化することが挙げられる。

## 文 献

- [1] J.R. Jackson, "Jobshop-like queueing systems," *Management Science*, vol.10, no.1, pp.131-142, Nov. 1963.
- [2] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red," In *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp.151-160, Aug. 2000.
- [3] A. Konrad, B.Y. Zhao, A.D. Joseph, and R. Ludwig, "A markov-based channel model algorithm for wireless networks," *Wireless Networks*, vol.9, no.3, pp.189-199, May. 2003.
- [4] K. Salamatian and S. Vaton, "Hidden markov modeling for network communication channels," *ACM SIGMETRICS Performance Evaluation Review*, vol.29, no.1, pp.92-101, Jun. 2001.
- [5] S. Chen, S.A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol.50, no.5, pp.1872-1896, Dec. 1989.
- [6] S.A. Billings and W.S.F. Voon, "Least squares parameter estimation algorithms for non-linear systems," *International Journal of Systems Science*, vol.15, no.6, pp.601-615, Apr. 1984.
- [7] P.J. Fleming and R.C. Purshous, "Evolutionary algorithms in control systems engineering: a survey," *Control Engineering Practice*, vol.10, no.11, pp.1223-1241, Nov. 2002.
- [8] Iba. H, Sato. T, and deGaris. H, "System identification approach to genetic programming," In *Proceedings of IEEE World Congress on Computational Intelligence*, pp.401-406, Jun. 1994.
- [9] C.K. Y, "Using genetic programming for developing relationship between engineering characteristics and customer requirements in new products," In *Proceedings of IEEE Conference on Industrial Electronics and Applications*, pp.526-531, Jun. 2011.
- [10] R. Behera, B. Pati, B. Panigrahi, and S. Misra, "An application of genetic programming for power system planning and operation," *ACEEE International Journal on Control System and Instrumentation*, vol.3, no.2, pp.15-20, Mar. 2012.
- [11] J.V. Hansen, "Genetic search methods in air traffic control," *Computers & Operations Research*, vol.31, no.3, pp.445-459, Mar. 2004.
- [12] A. Guven, "Linear genetic programming for time-series modelling of daily flow rate," *Journal of Earth System Science*, vol.118, no.2, pp.137-146, 2009.
- [13] J. Madar, J. Abonyi, and F. Szeifert, "Genetic programming for the identification of nonlinear input-output models," *Industrial & engineering chemistry research*, vol.44, no.9, pp.3178-3186, Mar. 2005.
- [14] G. Gary, J.Murray-Smith, David J., L. Yun, S.K. C., and W. Thomas, "Nonlinear model structure identification using genetic programming," *Control Engineering Practice*, vol.6, no.11, pp.1341-1352, Nov. 1998.
- [15] H. Iba, H. DeGaris, and T. Sato, "A numerical approach to genetic programming for system identification," *Evolutionary Computation*, vol.3, no.4, pp.417-452, Dec. 1995.
- [16] H.-T. Yang, C.-M. Huang, and C.-L. Huang, "Identification of armax model for short term load forecasting: an evolutionary programming approach," *IEEE Transactions on Power Systems*, vol.11, no.1, pp.403-408, Feb. 1996.
- [17] H. Okuyama, "System identification theory of the thermal network model and an application for multi-chamber airflow measurement," *Building and Environment*, vol.25, no.4, pp.349-363, 1990.
- [18] A. Orfila, J.M.E. Tapiador, and A. Ribagorda, "Evolving high-speed, easy-to-understand network intrusion detection rules with genetic programming," *Applications of Evolutionary Computing*, vol.5484, pp.93-98, 2009.
- [19] J.R. Koza, *Genetic Programming*, The MIT Press, 1995.
- [20] J.R. Koza, "Genetic programming," chapter 6, pp.79-119, The MIT Press, 1995.
- [21] L. Kleinrock, *Queueing Systems Volume1: Theory*, Jhon Wiley and Sons, 1975.