

Extremal Optimization による調停グラフの交差数減少

田村 慶一^{†1} 森 康真^{†1} 北上 始^{†1}

系統樹を比較する手法の1つとして調停が提案されている。調停では、系統樹を順序木と見なし、2つの順序木を向き合わせた調停グラフを作成する。調停を潤滑に進めるためには、順序木の葉節点間に交差がないか、もしくはできるだけ交差が少ない調停グラフを作成する必要がある。本論文では、Extremal Optimization (EO) による調停グラフの交差減少を提案する。EOを適用するにあたり、EOに対して、(1) ルーレット選択により状態遷移の候補となる構成要素を選択する、(2) 複数の近傍解中の最良解を次世代の解とする、という2つの改良を加えた。評価実験の結果、提案手法による調停グラフの交差数減少の方がEOと比較して優れていることを確認した。また、タブーサーチ、SAや遺伝的プログラミングを使用した手法よりも良い結果を求めることができることを確認できた。

Reducing Crossovers in Reconciliation Graphs with Extremal Optimization

KEIICHI TAMURA,^{†1} YASUMA MORI^{†1} and HAJIME KITAKAMI^{†1}

The reconciliation is a one of techniques for comparing phylogenetic trees. In the reconciliation, a phylogenetic tree is considered to be an ordered tree, order trees are made to be opposite, and the reconciliation graph is made. There is no intersection between leaf nodes to advance reconciliation to lubrication or it is necessary to make reconciliation graph where intersection is a little. This paper proposes the reducing crossovers in reconciliation graphs with Extremal Optimization (EO). There are two characteristics in the proposed method. The construct which serves as a candidate of a change state by roulette selecting is chosen. The best solution in two or more neighborhoods is chosen as the next solution. The experimental results show that the proposed method is excellent as compared with EO, tabu search, SA and GP.

1. はじめに

生物の多様性を解明するために、系統樹を比較する研究が行われている。たとえば、DNA塩基配列の違いから作成された分子進化系統樹は、遺伝子重複の現象が十分に反映されていない。そこで、生物分類樹と比較し、場合によっては、分子進化系統樹の見直しが行われる¹⁾。また、分子進化系統樹どうしを比較し、寄生と宿主の関係を明らかにする研究²⁾なども行われている。

系統樹を比較する手法の1つに調停³⁾⁻⁶⁾がある。調停では、生物分類樹と分子進化系統樹とをそれぞれ順序木と見なし、比較する2つの順序木を向き合わせたグラフを作成する。そして、作成したグラフのノードを1対1に対応付け、比較する2つの順序木の特徴や違いを明らかにしていく。以下、本論文では2つの順

序木を向き合わせたグラフのことを調停グラフと呼ぶこととする。

図1に調停グラフの簡単な例を示す。図1では、生物種‘a’、‘b’、‘c’、‘d’を別々の観点で分類した2つの順序木を図示している。図のように2つの順序木を向き合わせたグラフが調停グラフとなる。

調停を潤滑に進めるためには、順序木の葉ノード間に交差がないか、または、交差が少ない調停グラフを求める必要がある⁷⁾。図1の調停グラフにおいて同じ葉ノードどうしを線で結ぶと2つの交差が生じる。たとえば、図2のように、ノード‘2’の子ノード‘d’と子ノード‘1’の順序関係を入れ替えると、葉ノード間の交差数がゼロとなる。

このように、調停グラフの葉ノード間の交差数を削減することを調停グラフの交差数減少という。この問題は、組合せ最適化問題として定義することができ、厳密解を求めようとすると非常に大きな計算量を必要とする。たとえば、葉ノードの数を n とすると、葉ノードの並びの組合せは $n!$ 通り存在し、2つの順序

^{†1} 広島市立大学大学院情報科学研究科
Graduate School of Information Sciences, Hiroshima City University

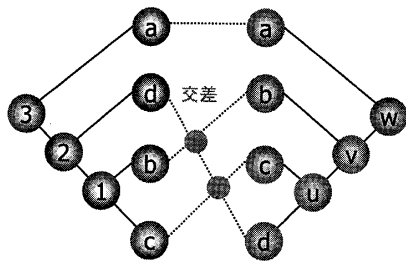


図1 調停グラフの例

Fig. 1 An example of reconciliation graph.

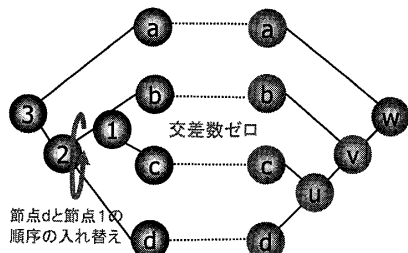


図2 交差数減少の例

Fig. 2 An example of reducing crossovers in reconciliation graph.

木の葉ノードの並びの組合せは $n! \times n!$ 通り存在する。これらの組合せの中で、交差数が最小になる調停グラフを求めることになる。

本論文では、発見的探索法の1つとして提案されている Extremal Optimization (EO)^{8),9)} による調停グラフの交差数減少を提案する。EOは、自然界において最適な構造が構築されるメカニズムに着目した発見的探索法である。EOを調停グラフの交差数減少に適用するにあたり、EOに対して、以下の2つの改良を加えた。改良点は以下のとおりである。

- (1) ルーレット選択により状態遷移の候補となる構成要素を選択：EOでは、適応度が最も小さいか、もしくは、適応度のランキングによるべき分布に従って（べき分布に従うEOを τ EOという）、構造中の構成要素が状態遷移の候補として選択される。調停グラフの交差数減少では、構成要素の適応度が、一定値以上大きくならない場合が考えられる。このような場合、適応度が最も小さい構成要素を状態遷移の候補として選択すると、局所解の周辺で探索の巡回が起こる可能性が高くなる。また、適応度のランキングによるべき分布に従って構成要素を選択すると、適応度の差が十分に反映されないという問題点がある。そこで、ルーレット選択を用いて状態遷移の候補となる構成要素を選択し、このような問題が発生するのを回避する。
- (2) 複数の近傍解の中から最良解を次世代の解として選択：EOと τ EOとでは、ランダムに近傍

解を1つ生成し、次世代の解とする。調停グラフの交差数減少では、近傍解によっては、解が急激に改悪の方向に進むことがある。そこで、解の急激な改悪を防ぐために、複数の近傍解から最良解を次世代の解とする。

提案手法を実際に実装し、評価実験を行った。評価実験の結果、EOや τ EOと比較して、提案手法の方が、より早い段階で最良な解を求めることができることを確認した。また、提案手法と、タブーサーチ、SAと遺伝的プログラミング(GP)による調停グラフの交差数減少を比較した結果、提案手法の方が局所解におちいらずに、より良い解を求めることができることを確認した。

本論文の構成は以下のとおりである。2章では、関連研究について述べる。3章では、調停グラフの交差数減少について、その問題定義を示す。4章では、Extremal Optimizationの概要を説明し、5章では、提案手法を示す。6章で評価実験の実験結果を示し、最後に、7章において、本論文をまとめる。

2. 関連研究

調停グラフの交差数減少に対する初期的なアプローチとして、葉ノードの位置関係を示す結合行列の行・列を探索的に入れ替える手法⁷⁾が提案されている。交差数がゼロとなる時、結合行列は単位行列となる。この手法では、このことに着目し、結合行列が単位行列になるように、結合行列の行・列を探索的に入れ替えていく。

結合行列の行・列を探索的に入れ替える手法は、局所探索の1つであり、単調な調停グラフであれば、高速に最適解を求めることができる。しかしながら、大域的な探索を行うことができないため、局所解におちいりやすいという問題点がある。

これに対して、遺伝的アルゴリズム(GA)¹⁰⁾を使用した手法¹¹⁾が提案されている。この手法は、Simple GAと局所探索とを組み合わせた手法（以下、GAによる手法と呼ぶ）である。Simple GAを使用して求めた最良解に対して、さらに、局所探索を行い、交差数を削減する。

GAによる手法は、大域的な探索が行えるため、結合行列の行・列を探索的に入れ替える手法と比較して、交差数を削減することに成功している。しかしながら、GAによる手法では、Simple GAを用いているため、解が初期集団に依存することと、探索の終盤で多様性が維持できないことが問題となっている。

そこで、世代交代モデルとしてMGG¹²⁾を採用し

た手法¹³⁾が検討されている。世代交代モデルとして MGG を使用しているため、課題となっていた、初期集合への依存と多様性の維持とを改善することができた。なお、この手法は、遺伝的プログラミング (GP) を使用して設計し直されている (以下、GP による手法と呼ぶ)。

しかしながら、GP による手法は、大規模な問題になると次の2つのことが問題となる。1つめの課題が処理時間で、2つめの課題が、パラメータ設定である。

GP による手法では、世代交代モデルとして MGG を使用しているため、解の収束が遅く、非常に多くの計算時間を必要とする。MGG は、ある程度、時間をかけても、多様性を維持することを目的とする手法ではあるが、調停グラフの規模が大きくなるほど、解の収束に時間がかかり、実用的な時間で問題を解くことができない。

また、パラメータ設定は、進化的計算において難しい問題ではあるが、GP による手法においても明確なパラメータ設定の指標がない。特に、大規模な問題になるほど、パラメータ設定の少しの差で、局所解に落ちいてしまうことが問題となっている。

EO は、スピングラス、グラフの分割問題、グラフの彩色問題、巡回セールスマン問題、センサデータのトラッキング、画像の対応付けに应用されるロバストな点对応付け、コミュニティ抽出問題などに应用されており、様々な分野で有効性が示されている¹⁴⁾⁻¹⁹⁾。

EO は、局所探索における近傍解生成に対して工夫を加えた探索手法であり、高速に探索を行うことができる。また、パラメータ設定が少ないということも EO の強みである。既存の応用研究では、大規模な問題の最適化に成功しており、EO の適用によって、調停グラフの交差数減少では、より評価の高い結果が得られると期待される。

3. 問題定義

本章では、調停グラフのデータ表現、その交差数の計算方法と、交差数減少とその制約について説明する。調停の生物学的な意義については、文献 7) を参照されたい。

3.1 データ表現

調停グラフ RG は、2つの順序木、順序木 $T1 = (V1, E1)$ 、順序木 $T2 = (V2, E2)$ から構成される。 $V1$ と $V2$ とはノード集合であり、 $E1$ と $E2$ とは辺集合である。

順序木は、それぞれ、1つのルートノード (根)、 $root1 \in V1$ と $root2 \in V2$ とを持つ。ルート

ノード以外のノードの入次数は1であり、出次数が0のノードを葉ノードと呼ぶ。また、ルートノードと葉ノード以外のノードを中間ノードと呼ぶこととする。

分類の対象となっている生物種のデータ数を n とすると、 $T1$ の葉ノード集合は $L1 = \{l1_1, l1_2, \dots, l1_n\}$ 、 $T2$ の葉ノード集合は $L2 = \{l2_1, l2_2, \dots, l2_n\}$ と表すことができる。ここで、各ノードのラベル名を返す関数名を \mathcal{L} とする。葉ノードのラベル名の集合を L_{leaf} とすると、

$$\mathcal{L}(l1_i) \in L_{leaf}, \mathcal{L}(l1_i) \neq \mathcal{L}(l1_j), i \neq j,$$

$$\mathcal{L}(l2_i) \in L_{leaf}, \mathcal{L}(l2_i) \neq \mathcal{L}(l2_j), i \neq j,$$

という関係が成り立っている。また、1つの順序木内では、同じラベル名を持つノードは存在しない。

2つの順序木を向き合わせたグラフが調停グラフとなる。そして、同じラベル名を持つ葉ノードを線で結んだときに、その線上にできる交差の数が調停グラフの交差数である。

次に、ある辺 $(v, w) (\in E1, \text{もしくは}, \in E2)$ を考える。ノード v は、ノード w の親ノードであり、ノード w は、ノード v の子ノードとなる。共通の、親ノードを持つノードは兄弟であり、兄弟であるノード間には順序関係が存在する。

図1では、順序関係を上下で図示している。たとえば、ノード '2' の兄弟はノード 'd' とノード '1' である。この例では、ノード 'd' の方がノード '1' よりも順序が高くなっている。

葉ノードからルートノードへは必ず1つの道が存在し、その道に存在するノードを、当該葉ノードの先祖ノードと呼ぶ。以下、本論文では、 $T1$ のある葉ノード $l1_i$ のすべての先祖ノード集合を、 $AS(T1, l1_i)$ 、 $T2$ のある葉ノード $l2_i$ のすべての先祖ノード集合を、 $AS(T2, l2_i)$ と表すものとする。

3.2 結合行列と交差数

葉ノードの順序は、それぞれの順序木を深さ優先探索で探索することで取り出すことができる。順序木における葉ノードの順序が次のような配列として与えられたとする。

$$OL1 = [ol1_1, ol1_2, \dots, ol1_n]$$

$$(ol1_i \in L1, \mathcal{L}(ol1_i) \in L_{leaf})$$

$$OL2 = [ol2_1, ol2_2, \dots, ol2_n]$$

$$(ol2_i \in L2, \mathcal{L}(ol2_i) \in L_{leaf})$$

このとき、結合行列 M の要素 $m_{i,j}$ を以下のように、

$$m_{i,j} = \begin{cases} 1 & \text{if } \mathcal{L}(ol1_i) = \mathcal{L}(ol2_j), \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

と表すと、葉ノード間の交差数は、以下の式で求める

ことができる。

$$\sum m_{j,\beta} m_{k,\alpha} [1 \leq j < k \leq n, 1 \leq \alpha < \beta \leq n] \quad (2)$$

3.3 交差数減少とその制約

葉ノードの順序を変えることで交差数を削減することができる。ただし、交差数減少では、以下の制約を守る必要がある。

- (1) 順序木のノード間で交差を発生させてはならない。
- (2) 順序木が示す分子進化系統樹の系統や生物分類樹の分類を壊してはならない。

上述の制約を守るためには、中間ノードにおいて、その子ノードの順序関係を変更することで、葉ノードの順序を変化させる必要がある。

たとえば、図1において、葉ノード'd'と葉ノード'c'とを入れ替えたい場合、単に、葉ノード'd'と葉ノード'c'とを入れ替えると順序木そのものに交差が生じてしまう。この場合、葉ノード'd'と葉ノード'c'の共通の先祖であるノード'2'に着目して、その兄弟であるノード'1'と葉ノード'd'の順序関係を入れ替えることで、順番の入れ替えを行う必要がある。

4. Extremal Optimization

EOでは、複数の個体間で自然淘汰を行うのではなく、個体を構成する複数の構成要素間で自然淘汰を行う。つまり、個体を構成する各構成要素の適応度を向上させていくことで、最適な個体を求めていく。

具体的には、適応度が最も小さい構成要素を選び、選択した構成要素を状態遷移させることで探索を進めていく。適応度が最も小さい構成要素は、状態遷移にともない適応度が大きくなる可能性が高い。構成要素の状態遷移にともない、その周辺にある構成要素も影響を受け変化する。この結果、周辺にある構成要素の適応度が小さくなったとしても、いずれ、その構成要素も状態遷移の候補として選ばれるため、全体として、各構成要素の適応度が大きくなっていくと期待される。そして、各構成要素の適応度が大きくなることで、個体の評価も高くなり、最適な個体を作成していくことができる。

EOのアルゴリズムをAlgorithm 1に示す。実行可能解 S は、 $S = \{O_1, O_2, \dots, O_n\}$ のように、 n 個の構成要素から構成され、構成要素 O_i の適応度を λ_i と表すものとする。また、 \mathcal{F} は個体の適応度を返す関数であり、この値が大きいほど適応度が大きい。初期解の生成方法、構成要素とその適応度の計算方法や状

態遷移の方法については、アプリケーションごとに異なるため、EO自体に明確な方法は示されていない。

最初に、適応度 λ_i が最も小さい構成要素 O_i を状態遷移の候補として選択する。そして、当該構成要素を対象として、 S に対して状態遷移を行った近傍解をランダムに生成し、次世代の解とする。以下、構成要素の選択と状態遷移を繰り返しながら、解の探索を進めていく。

Algorithm 1 EO_BASIC

- 1: 初期解 S_{init} を生成。
 - 2: $S_{best} := S_{init}$; /* S_{best} は最良解 */
 - 3: $S_{current} := S_{init}$; /* $S_{current}$ は現在の解 */
 - 4: **while** 指定した世代数以下ならば **do**
 - 5: $S_{current}$ のすべての構成要素 O_i に関して、その適応度 λ_i を算出。
 - 6: 適応度 λ_i が最も小さい構成要素 O_i を選択。
 - 7: 構成要素 O_i を対象として、 $S_{current}$ に対して状態遷移を行った近傍解 S_{next} を生成。
 - 8: $S_{current} := S_{next}$;
 - 9: もし、 $\mathcal{F}(S_{current}) > \mathcal{F}(S_{best})$ ならば、
 $S_{best} := S_{current}$;
 - 10: **end while**
-

EOでは、適応度 λ_i が最も小さい構成要素を選択する。しかしながら、状態遷移によっては、最も勾配の高い尾根に登るだけの探索になり、局所解におちいる可能性がある。

そこで、この問題を解決するために、 τ EOが提案されている。 τ EOでは、構成要素の選択を確率的に行う。まず、構成要素を適応度 λ_i でランク付けする。そして、適応度がランク n である構成要素に対して、 $n^{-\tau}$ に比例した選択率を与える。そして、この選択率によって状態遷移を行う構成要素を選択する。これにより、選択に対してランダム性が加わり、局所解におちいるのを防ぐことができる。

5. 提案手法

調停グラフの交差数減少を手作業で行う場合、交差を多く作っている葉ノードに着目して交差数を削減すると、交差数をうまく削減できることが多い。つまり、調停グラフの交差数減少では、交差数を多く作っており、全体の評価を下げている悪い部分を積極的に改善しようとする、より良い解を求めることができる。

このことは、悪い部品を積極的に改善しようとする

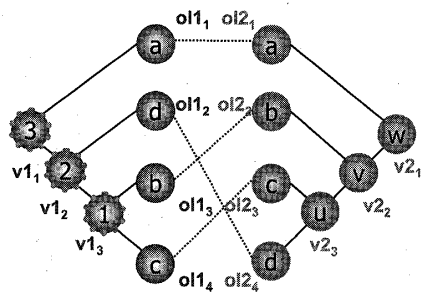


図 3 構成要素

Fig. 3 Constituent factors.

EO の考えに似ている。よって、EO を調停グラフの交差数減少に対して適用することで、他の手法と比較して、より良い解を求めることができる可能性がある。ただし、5.4 節でも述べるが、EO を適用するにあたり、調停グラフの交差数減少特有の問題点があるため、EO に対して 2 つの改良を加えた。

以下、EO を適用するにあたり必要となる、構成要素の定義、適応度の計算方法と状態遷移の仕方を説明し、改良点とそのアルゴリズムを示す。

5.1 構成要素

構成要素を、次のように、同じラベル名を持つ葉ノードのペアとする。

$$O_i = \{ol1_i, ol2_{\delta(i)}\} \quad (\mathcal{L}(ol1_i) = \mathcal{L}(ol2_{\delta(i)}))$$

関数 $\delta(i)$ は、葉ノードの順序を示す配列 $OL1$ の要素 $ol1_i$ のラベル名と同じラベル名を持つ $OL2$ の要素の添字番号を返す関数とする。

たとえば、図 3 の調停グラフを例として考える。ラベル名が同じである葉ノードを考えると、 $\delta(i)$ が返す値は、それぞれ、 $\delta(1) = 1$ 、 $\delta(2) = 4$ 、 $\delta(3) = 2$ 、 $\delta(4) = 3$ となる。また、構成要素は、それぞれ、 $O_1 = \{ol1_1, ol2_1\}$ 、 $O_2 = \{ol1_2, ol2_4\}$ 、 $O_3 = \{ol1_3, ol2_2\}$ 、 $O_4 = \{ol1_4, ol2_3\}$ となる。

葉ノードのみを構成要素として考えるのは、適応度となる交差数の算出においては、葉ノードの並びのみ考えればよいためである。

5.2 適応度

構成要素 $O_i = \{ol1_i, ol2_{\delta(i)}\}$ の適応度は、葉ノード $ol1_i$ と葉ノード $ol2_{\delta(i)}$ の交差数を 2 で割った値 C_i を使用して、

$$\lambda_i = \frac{C - C_i}{C}, \quad (3)$$

と定義する (ただし、 $C = 0$ の場合を除く)。 C は全体の交差数であり、

$$C(M) = \sum_{i=1}^n C_i, \quad (4)$$

が成り立っている。

C_i は以下の計算式で求めることができる。

$$C_i = \sum_{l=i+1}^n \sum_{m=1}^{\delta(i)-1} \frac{ml_m}{2} + \sum_{l=1}^{i-1} \sum_{m=\delta(i)+1}^n \frac{ml_m}{2} \quad (5)$$

図 3 の例だと、 $\lambda_1 = 1$ 、 $\lambda_2 = 1/2$ 、 $\lambda_3 = 3/4$ 、 $\lambda_4 = 3/4$ となる。交差数が生じていない構成要素 O_1 の適応度が最も大きく 1 となっている。

5.3 状態遷移

状態遷移は、中間ノードにある兄弟ノードの順序関係を入れ替えることで行う。以下に、構成要素 $O_i = \{ol1_i, ol2_j\}$ ($j = \delta(i)$) の状態遷移の手順を示す。構成要素 $O_i = \{ol1_i, ol2_j\}$ ($j = \delta(i)$) の選択方法については、次節で説明する。

- (1) 順序木 $T1$ 、順序木 $T2$ をランダムに 1 つ選択する。
- (2) 順序木 $T1$ が選択されたならば、 $AS(T1, ol1_i)$ を候補集合 CS とする。順序木 $T2$ が選択されたならば、 $AS(T2, ol2_j)$ を候補集合 CS とする。
- (3) 候補集合 CS の中から子ノードの数が 2 つ以上の中間ノードをランダムに 1 つ選択する。
- (4) 選択したノードの子ノードの中から 2 つの兄弟をランダムに選択し、順序関係を入れ替える。

図 3 で示す調停グラフの構成要素 $O_4 = \{ol1_4, ol2_3\}$ が状態遷移の対象として選ばれた場合を考える。もし、順序木 $T1$ が選択されたとすると、点線で示すノードが候補集合 CS となる。候補集合からノードをランダムに選び、その兄弟を入れ替えることで状態遷移を行う。たとえば、候補集合からノード $v1_1$ が選ばれた場合、ノード $v1_2$ とノード $ol1_1$ の順序関係を入れ替えることで状態遷移が行われる。

状態遷移では、順序関係を入れ替えの対象となったノードにより、局所的な変化が起こる場合と、大域的な変化が起こる場合が考えられる。入れ替えの対象となるノードをルートノードとする部分木に存在する葉ノードの数が多きとき、大域的な変化が起こる。逆に、数が少ないときは、局所的な変化が起こる。ただし、葉ノードの数が少ない場合でも、入れ替える葉ノード間の距離が大きい場合は、大域的な変化が起こる。

5.4 アルゴリズム

Algorithm 2 に、提案手法の流れを示す。 \mathcal{F} は個体の適応度を返す関数であり、この値が大きいのほど適応度が大きいものとする。まず、ルーレット選択により、構成要素を 1 つ選択する。ルーレット選択では、 $1/\lambda_i$

に比例した選択率を与える。次に、前節で示した状態遷移の方法で、選択した構成要素を状態遷移することで近傍解を1つ生成する。そして、指定した近傍解の生成数に達するまで、ルーレット選択と近傍解の生成を繰り返す。指定した近傍解の生成数に達すると、生成した近傍解の中から最良解を選択し、次世代の解とする。

Algorithm 2 EO-CROSS

- 1: 初期解 S_{init} として、ランダムに調停グラフを生成。
 - 2: $S_{best} := S_{init}$;
 - 3: $S_{current} := S_{init}$;
 - 4: **while** 指定した世代数以下ならば **do**
 - 5: $S_{current}$ のすべての構成要素 O_i に関して、適応度 λ_i を算出。
 - 6: $N := \phi$;
 - 7: **while** 指定した近傍解生成数以下ならば **do**
 - 8: ルーレット選択により、構成要素 O_i を1つ選択。
 - 9: 構成要素 O_i を対象として、 $S_{current}$ に対して状態遷移を行った近傍解 $N(O_i)$ を生成。
 - 10: $N := N \cup N(O_i)$;
 - 11: **end while**
 - 12: N の中から最良解 N_{best} を選択し、次世代の解 S_{next} とする。
 - 13: $S_{current} := S_{next}$;
 - 14: もし、 $\mathcal{F}(S_{current}) > \mathcal{F}(S_{best})$ ならば、
 $S_{best} := S_{current}$ 。
 - 15: **end while**
-

提案手法では、EO に対して、(1) ルーレット選択、(2) 複数近傍解の生成、という2つの改良を加えている。以下、これらの改良を加えた理由を述べる。

調停グラフの交差数減少では、構成要素の適応度が一定値以上大きくならない場合が考えられる。つまり、交差数をこれ以上下げられない構成要素が存在する。これは、最適解が得られたとしても、交差数がある一定数以上減らすことができない部分木が存在するためである。

このような場合、EO による構成要素の選択方法だと、局所解の周辺で探索の巡回が起こる可能性が高くなる。たとえば、適応度をこれ以上向上させることができない、適応度が最も小さい構成要素があった場合、当該構成要素が状態遷移の候補として必ず選択される。当該構成要素が状態遷移を起こしたとしても、改悪の

方向にしか解が遷移しない。そこで、いったん解が改悪の方向に動いたとしても、また、当該構成要素が選択され、解が改善されるといったことを繰り返すようになる。

一方、 τ EO では、 τ の値が大きいときは、適応度の小さな構成要素が選ばれやすく、EO と同じ問題が発生する。また、 τ の値を小さくすると、適応度が大きな構成要素が選択される確率も高くなる。しかしながら、適応度の差が反映されないため、適応度の大きな構成要素の選択率が高くなってしまう。

たとえば、適応度が、それぞれ、1, 1/2, 3/4, 3/4 となる構成要素があった場合と、適応度が、それぞれ、7/8, 5/8, 3/4, 3/4 となる構成要素があった場合とを考える。ランキングに従うと、2つの場合ともに同じランキングが付与されるため、結果的に同じ選択率が付与される。

このように、 τ EO では、適応度に差があったとしても、ランキングに変換されるため、その差が考慮されなくなる。これにより、場合によっては、適応度の大きな構成要素に大きな選択率が与えられることになる。探索の序盤で、適応度の大きな構成要素に大きな選択率が与えられるとランダムサーチに近くなり、解の探索がなかなか進まなくなる。

そこで、ルーレット選択を用いて、適応度の大きい構成要素も、適応度に応じて、状態遷移の候補として選ばれやすくする。ルーレット選択では、 $1/\lambda_i$ に比例した選択率を付与するために、適応度の差が選択率に反映され、探索の序盤で、解の探索がなかなか進まなくなることを回避することができる。

EO や τ EO では、ランダムに近傍解を1つ生成し、次世代の解としている。調停グラフの交差数減少では、局所的な操作により、局所的な変化が発生することや大域的な変化が発生することがある。どちらの変化が起こるかは、実際に操作しなければ分からない。大域的な変化では、急激に改善の方向に進むことや、改悪の方向に進むことがある。

そこで、解が急激に改悪の方向に進まないようにするために、近傍解を複数生成して、複数の近傍解の中から最良の解を次世代の解とする。これにより、複数生成した近傍解の中に改悪のものがあったとしても次世代の解として選ばれにくくなり、また、大幅に改善されたものがあれば、それが次世代の解として選ばれるようになる。

表 1 データセット
Table 1 Data set.

データ名	生物分類樹		分子進化系統樹		最小交差数
	非葉ノード数	葉ノード数	非葉ノード数	葉ノード数	
hashimoto97	201	40	39	40	9
コケ	83	207	187	207	-

6. 評価実験

6.1 実験概要

提案手法の有効性を示すために、評価実験として、次の4つの実験を行う。

実験1では、EO、もしくは、 τ EOによる調停グラフの交差数減少と、提案手法による調停グラフの交差数減少とを比較する。実験2では、提案手法における複数近傍解の生成とルーレット選択との有効性について検証を行う。実験3では、タブーサーチ、もしくは、SAによる調停グラフの交差数減少と、提案手法による調停グラフの交差数減少とを比較する。実験4では、文献13)で提案されているGPによる手法と提案手法とを比較する。

表1に実験に使用したデータセットの内容を示す。データセット「hashimoto97」はハウスキーピング遺伝子による分子進化系統樹とその生物分類樹から構成される。このデータセットの最適解の交差数は9であることが分かっている。データセット「コケ」は、207種のコケの生物分類樹と rps4 遺伝子による分子進化系統樹とで構成される。

6.2 実験 1

図4(a)と図5(a)とに、データセット「hashimoto97」を使用したときの、世代数と最小交差数(最良解の交差数)、経過時間と最小交差数の測定結果をそれぞれ示す。提案手法における近傍解の生成数は50である。近傍解の生成数が50のときが最も良い結果が得られているため、この生成数を比較対象とする。また、 τ EOにおける τ は1.5とする。この τ の値は、先行研究において τ EOにおいて、最も良い値として推奨されている値であり、今回の測定においても、最も良い結果が得られた値であった。また、測定は、5回行い、最良の3つの結果の平均を測定値として示す。

EOと τ EOとは、最小交差数は100より下がることはないが、提案手法では、最適解である交差数9を求めることができた。近傍解を1つしか生成しないEOと τ EOとでは、100前後で、解が改悪になったり改善したりと、堂々巡りをしていると考えられる。

提案手法は、複数の近傍解を生成するために、EOと、 τ EOと比較して処理時間が大きい。図5(a)を見

ると、最初、単位時間あたりに処理できる世代数が多いEOと τ EOの方が交差数をより減少させることができている。しかしながら、これはすぐに逆転し、処理時間の点においても、提案手法の方が優れた結果を示している。

同様に、図4(b)と図5(b)とに、データセット「コケ」を使用したときの、世代数と最小交差数、処理時間と最小交差数(最良解の交差数)の測定結果を示す。測定条件は、データセット「hashimoto97」のときと同じである。データセット「コケ」においても、データセット「hashimoto97」同様に、提案手法の方が良い結果を示している。「コケ」は「hashimoto97」と比較して規模が大きな問題であるため、その差は非常に大きなものとなっている。

6.3 実験 2

図6(a)と図6(b)とに、提案手法において、近傍解の生成数を変化させたときの測定結果を示す。測定条件は実験1と同じであるが、それぞれ、50世代、1000世代までを拡大して図示している。

近傍解の生成数が10と他の生成数とを比較すると、他の生成数の方が、より早い段階で交差数を削減できている。複数の近傍解を生成することで、性能が向上しており、複数の近傍解を作成することは、調停グラフの交差数減少において有効であるといえる。

次に、ルーレット選択の効果を検証するために、提案手法においてルーレット選択を行っている部分を τ EOにおける選択方法に置き換えたものを実装し、提案手法と比較する。

図7(a)と図7(b)とに、データセット別に測定結果を示す。測定条件は実験1と同じである。 τ EOによる選択では、最も良い結果が得られている $\tau=1.5$ と、 $\tau=0.5$ と値を小さくして、適応度の大きい構成要素に対して選択率が高くなるような場合とを測定した。

データセット「hashimoto97」に関しては、ルーレット選択と τ EOによる選択($\tau=1.5$)とはともに、ほぼ同じ結果が得られた。しかしながら、ルーレット選択は、最適解である最小交差数9を求めることができているが、 τ EOによる選択($\tau=1.5$)は、最適解である最小交差数9を求めることができていない。これは、 τ の値が大きいために、最後の1つの交差を削

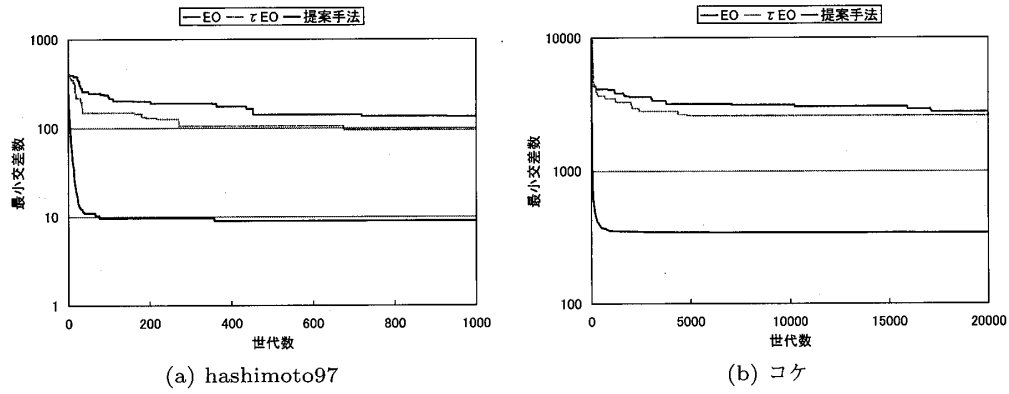


図 4 世代数と最小交差数
Fig. 4 The number of the minimum intersections (generations).

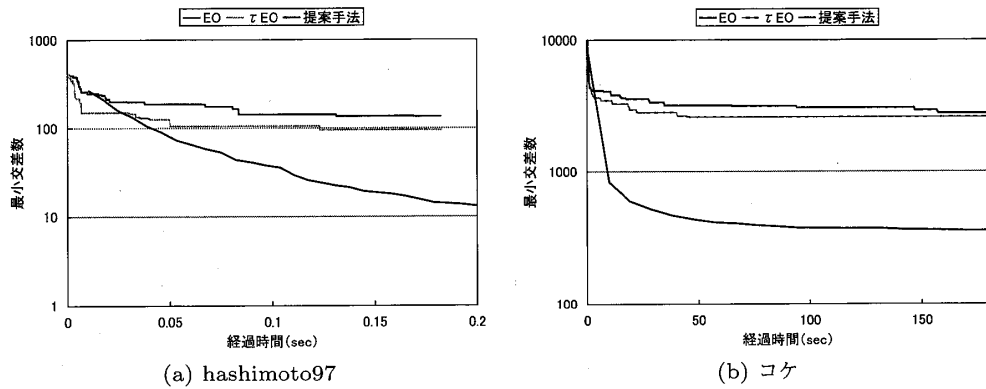


図 5 経過時間と最小交差数
Fig. 5 The number of the minimum intersections (elapsed time).

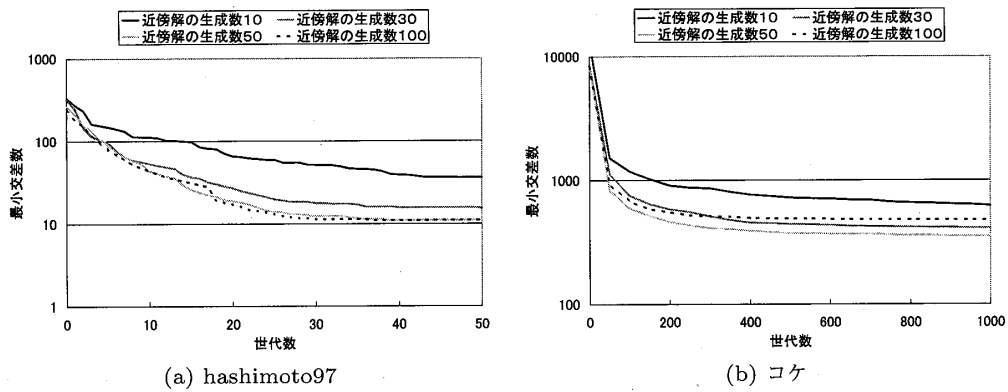


図 6 近傍解生成数と最小交差数
Fig. 6 The number of the minimum intersections (the number of neighborhoods).

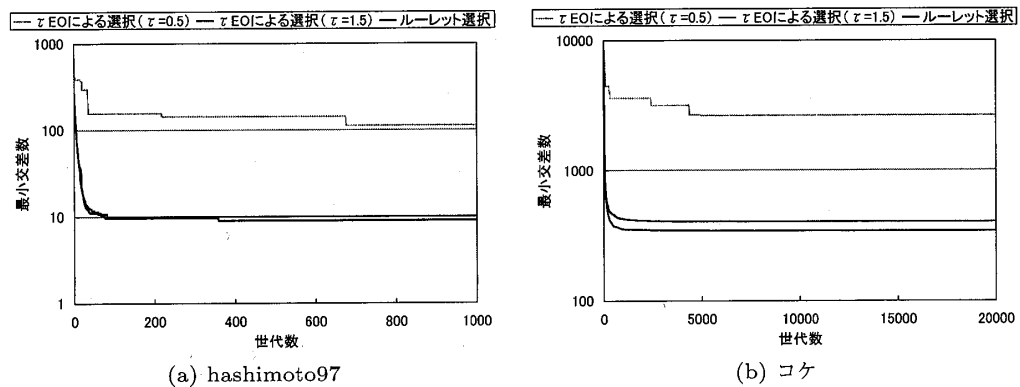


図 7 ルーレット選択と τEO による選択との比較
Fig. 7 A compare of roulette selection with τEO's selection.

減するとき、1つの交差を持つ構成要素よりも、交差数が多い構成要素が選ばれやすくなっているためである。

また、 τ の値を0.5とした場合では、 τ が1.5のときよりも、適応度の大きな構成要素が選択されやすくなるが、5.4節で述べたように初期の探索の収束が遅く、交差数をうまく削減することができていない。

データセット「コケ」に関しては、 τ EOによる選択($\tau=1.5$)では局所解におちいっており、ルーレット選択は τ EOによる選択($\tau=1.5$)よりも交差数を削減することができた。「hashimoto97」よりも交差数を削減できていないのは、交差数をこれ以上削減できない構成要素が「hashimoto97」よりも多いためである。よって、問題の規模が大きくなるほど、ルーレット選択と τ EOによる選択との差は大きくなると考えられる。また、 τ の値を0.5に設定しても、「hashimoto97」と同様に、初期の探索の収束が遅く、交差数をうまく削減することができていない。

以上の結果から、ルーレット選択による構成要素の選択と複数の近傍解を生成する手法は、調停グラフの交差数減少において効果があるといえる。

6.4 実験 3

実験3では、測定を200回行い、各測定の最後に得られた最小交差数の度数分布を測定する。最大世代数は5000回で測定を行う。提案手法とタブーサーチにおける近傍解の生成数は50する。タブーサーチは文献20)で示されているアルゴリズムを用いる。SAは基本的なアルゴリズムを使用し、パラメータである初期温度は1000、温度係数を0.9とした。SAのパラメータは、最も良い結果が得られている値を用いた。

タブーサーチとSAにおける近傍解の生成はランダムに行う。つまり、ランダムに構成要素を選択して状態遷移の候補とする。提案手法は近傍解の生成方法に特徴があり、一方、SAやタブーサーチは、新しい解を受容するか否かの決定方法に特徴がある。この実験では、この2つの比較を行うことになる。

図8に、データセット「hashimoto97」を使用したときの測定結果を示す。グラフは、棒グラフが頻度を示し、線が累積分布を示している。

提案手法、タブーサーチとSAともに最適解を求めることができていたが、最も頻度が大きいのが提案手法である。また、タブーサーチとSAには、最小交差数30周辺にいくつかの山が存在する。これは、最小交差数が30の周辺に局所解が存在するためである。この30前後の山は、提案手法では発生していない。

提案手法では、近傍解を複数生成する。調停グラフ

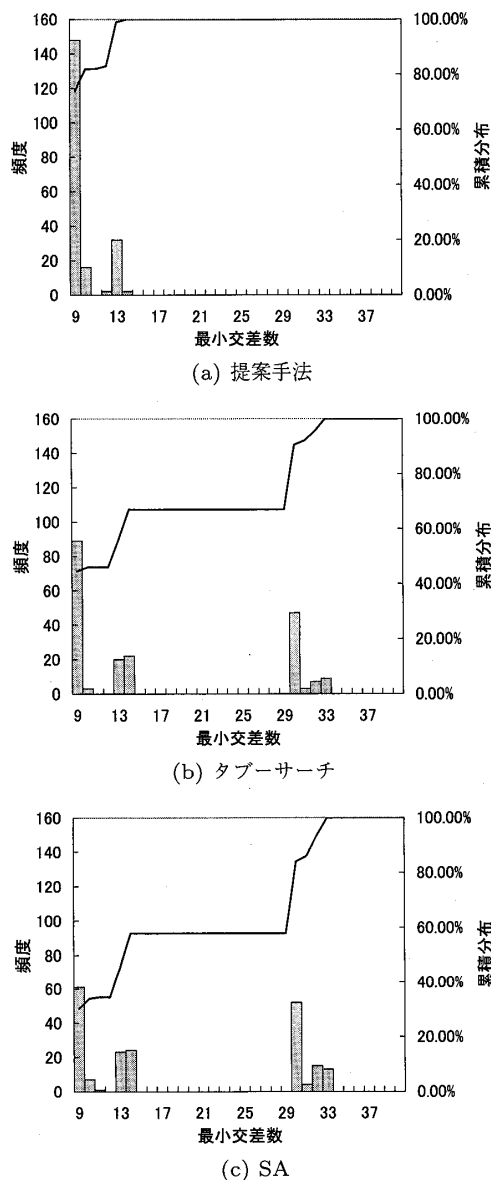
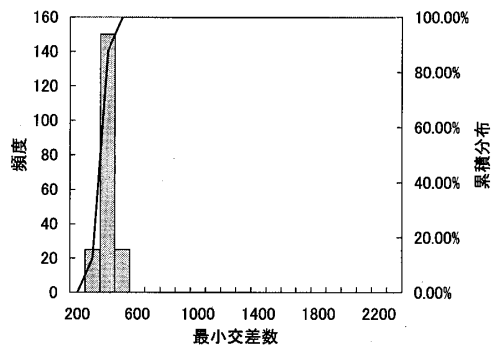


図8 最小交差数の度数分布 (hashimoto97)
Fig. 8 Frequency distribution of the number of the minimum intersections (hashimoto97).

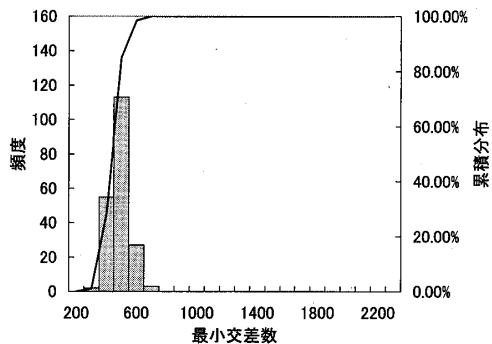
の交差数減少の特徴により、生成した近傍解の中には大域的な変化でかつ解が改善されるような近傍解も存在する。このような特徴により、提案手法では局所解におちいるのを回避できているといえる。

図9に、データセット「コケ」を使用したときの測定結果を示す。グラフは、棒グラフが頻度を示し、線が累積分布を示している。この結果は、データセット「hashimoto97」と比較して差が出ている。

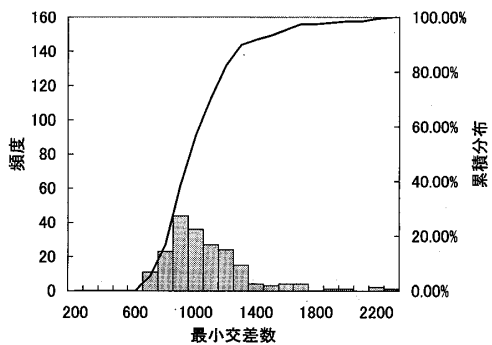
まず、SAは、タブーサーチと提案手法と比べて、求められた最小交差数はかなり大きなものとなっている。今回は、単純なSAを用いたため、再アニーリングなどの改良を加えたらさらによい結果を求めることができると期待される。しかし、さらにパラメータ設定が増え、パラメータは問題や問題の規模に依存するため、



(a) 提案手法



(b) タブーサーチ



(c) SA

図9 最小交差数の度数分布 (コケ)

Fig. 9 Frequency distribution of the number of the minimum intersections (moss).

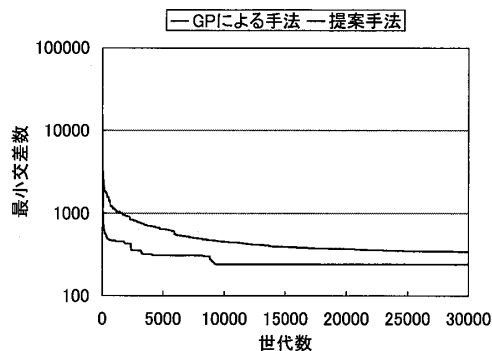
SA が最適な解法であるとはいえない。

次に、累積分布を見ると、提案手法では、最小交差数 400 以下が 85% で、タブーサーチでは、最小交差数 500 以下が 85% となっている。提案手法では、最小交差数が 300 以下の結果が 20 回以上得られているが、タブーサーチでは 1 回であった。

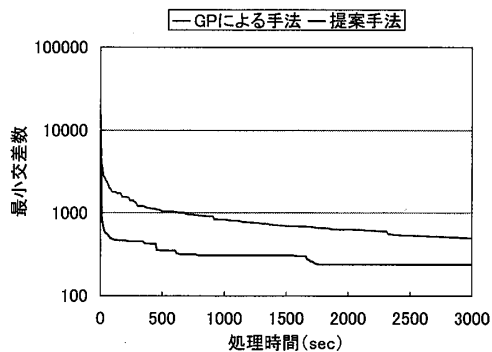
6.5 実験 4

図 10 に遺伝的プログラミング (GP) による手法と提案手法とによる調停グラフの交差数減少の比較結果を示す。

この測定では、データセットとしては「コケ」のみを用いる。GP のパラメータは、データセット「コケ」に対しても最も良い結果が得られるものを使用し、個体数 100、子個体の生成数 100、突然変異率 1% で測



(a) 世代数と最小交差数



(b) 処理時間と最小交差数

図 10 GP による手法との比較

Fig. 10 The compare with the genetic programming.

定を行った。また、提案手法における近傍解生成数は、GP の個体数と同じ 100 とする。

図 10 (a) は、世代数ごとの最小交差数を示している。提案手法では、世代数の早い段階で、交差数が 1000 以下となっている。また、世代数が 10000 の段階で、交差数が 240 の調停グラフが得られ、それから変化が起こっていない。

一方、GP による手法では、世代交代モデル MGG を使用しているため、なだらかに交差数の削減が進んでいる。世代数が 30000 の段階において、提案手法はすでに収束をしているが、GP による手法では削減が続いている。

しかしながら、さらに、世代数を 300 万まで進めていくと GP による手法では、交差数が 300 で解が収束し、提案手法と同じ結果を得ることができなかった。これは、収束が進み、すべての個体が似た構造となり、個体の多様性がなくなったためである。

図 10 (b) は、処理時間と最小交差数の関係を示したグラフである。最初の 3,000 秒までを拡大して示している。提案手法では、1 秒以内で交差数が 1000 以下となっているが、GP による手法では、700 秒でやっと交差数が 1000 以下となっている。

個体数が 1 つの提案手法と複数の個体を持つ GP を一律に比較はできないが、個体数が 1 つであっても

GP による手法よりも確実に探索を進めていくことができる提案手法の方が優れているといえる。また、提案手法の方が、より実用的な時間で問題を解くことができると考えられる。

7. ま と め

本研究では、EO による調停グラフの交差数減少を提案した。提案手法では、EO をそのまま適用するのではなく、ルーレット選択と複数近傍解生成という 2 つの改良を加えた。評価実験の結果、提案手法は、EO と τ EO と比べて、良い解を求めることができた。また、提案手法は、他の発見的探索手法であるタブーサーチ、SA や GP と比較して評価の高い結果を示した。

これからの課題として、最適な近傍解生成数の自動設定や局所解から動的な脱出方法の検討を行う予定である。また、他の組合せ最適化問題に適用して、提案手法自体の一般性を検証していきたい。

謝辞 本研究の一部は、日本学術振興会・科学研究費補助金（基盤研究（C）,（一般）, 課題番号：17500097）、文部科学省・科学研究費補助金（課題番号：18700094）の支援により行われた。

参 考 文 献

- 1) Stege, U.: Gene Trees and Species Trees: The Gene-Duplication Problem in Fixed-Parameter Tractable, *Workshop on Algorithms and Data Structures*, pp.288–293 (1999).
- 2) Page, R.: Temporal congruence revisited: Comparison of mitochondrial DNA sequence-divergence in cospeciating pocket gophers and their chewing lice, *Systematic Biology*, Vol.45, pp.151–167 (1996).
- 3) Goodman, M., Czelusniak, J., Moore, G., Romero-Herrera, A. and Matsuda, G.: Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences, *Systematic Zoology*, Vol.28, pp.132–163 (1979).
- 4) Page, R.: Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas, *Systematic Biology*, Vol.43, pp.58–77 (1994).
- 5) Page, R. and Charleston, M.: From gene to organismal phylogeny: Reconciled trees and the gene tree species tree problem, *Molecular Phylogenetics and Evolution*, Vol.7, No.2, pp.231–240 (1997).
- 6) Ma, B., Li, M. and Zhang, L.: From Gene Trees to Species Trees, *SIAM Journal on Computing*, Vol.30, No.3, pp.729–752 (2001).
- 7) 北上 始, 森 康真, 太田聡史, 斎藤成也: 異種系統樹間の調停のためのゼロ交差制約の充足, 情報処理学会論文誌: データベース, Vol.40, No.5, pp.1–14 (1999).
- 8) Boettcher, S. and Percus, A.: Nature's way of optimizing, *Artificial Intelligence*, Vol.119, No.1-2, pp.275–286 (2000).
- 9) Boettcher, S.: Extremal optimization: heuristics via coevolutionary avalanches, *Computing in Science and Engineering*, Vol.2, No.6, pp.75–82 (2000).
- 10) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Professional (1989).
- 11) Kitakami, H. and Mori, Y.: Reducing Crossovers in Reconciliation Graphs using the Coupling Cluster Exchange Method with a Genetic Algorithm, *Active Mining, IOS press*, Vol.79, pp.163–174 (2002).
- 12) 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, 人工知能学会誌, Vol.12, No.5, pp.734–744 (1996).
- 13) 札埜有香, 田村慶一, 森 康真, 北上 始: 遺伝的プログラミングによる調停グラフの交差数減少とその並列分散処理, 第 16 回データ工学ワークショップ (DEWS2005), 電子情報通信学会データ工学研究専門委員会, Online Proceedings (2005).
- 14) Boettcher, S. and Percus, A.G.: Extremal Optimization: Methods derived from Co-Evolution, *Proc. Genetic and Evolutionary Computation Conference*, pp.825–832 (1999).
- 15) Meshoul, S. and Batouche, M.: Robust Point Correspondence for Image Registration Using Optimization with Extremal Dynamics, *Proc. 24th DAGM Symposium on Pattern Recognition*, London, UK, pp.330–337, Springer-Verlag (2002).
- 16) Boettcher, S. and Percus, A.G.: Extremal Optimization at the Phase Transition of the 3-Coloring Problem, *Physical Review E*, Vol.69, p.066703 (2004).
- 17) Svenson, P.: Extremal optimization for sensor report pre-processing, *PROC SPIE*, Vol.5429, pp.162–171 (2004).
- 18) Zhou, T., Bai, W.-J., Cheng, L.-J. and Wang, B.-H.: Continuous extremal optimization for Lennard-Jones Clusters, *Physical Review E*, Vol.72, p.016702 (2005).
- 19) Duch, J. and Arenas, A.: Community detection in complex networks using Extremal Optimization, *Physical Review E*, Vol.72, p.027104 (2005).
- 20) Misevicius, A.: A tabu search algorithm for the quadratic assignment problem, *Comput.*

Optim. Appl., Vol.30, No.1, pp.95-111 (2005).

(平成 19 年 8 月 6 日受付)

(平成 19 年 9 月 24 日再受付)

(平成 19 年 10 月 26 日再受付 (2))

(平成 19 年 11 月 11 日採録)



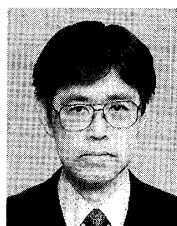
田村 慶一 (正会員)

1998 年九州大学工学部情報工学科卒業。2000 年同大学大学院システム情報科学研究科知能システム学専攻修士課程修了。2003 年同大学院システム情報化学府知能システム学専攻博士後期課程単位取得のうえ満期退学。博士 (情報科学)。現在広島市立大学大学院情報科学研究科助教。データベースとその並列分散処理に関する研究に従事。日本データベース学会, IEEE CS 各会員。



森 康真 (正会員)

1994 年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了。現在広島市立大学大学院情報科学研究科助教。データマイニングの研究に従事。電子情報通信学会, 人工知能学会, 日本データベース学会, IEEE CS, ACM 各会員。



北上 始 (正会員)

1976 年東北大学大学院工学研究科博士前期課程修了。同年富士通株式会社入社。以後、富士通研究所、新世代コンピュータ技術開発機構、国立遺伝学研究所客員助教授を経て、1994 年広島市立大学情報科学部教授、現在に至る。データマイニング、生命情報学、知識ベース等の教育研究に従事。博士 (工学)。情報処理学会 25 周年記念論文、日本工学教育協会論文論説賞、情報処理学会一般情報処理教育委員会委員、人工知能学会評議員、電子情報通信学会データ工学ワークショップ DEWS2008 組織委員、日本データベース学会 BI 研究グループ運営委員、情報処理学会 (TOM) 論文誌編集委員、IEEE, ACM, 電子情報通信学会各会員。