

# 特徴的なブロック保存型外平面的グラフパターンの獲得における ブロック木パターンの深さラベル列の利用

Using Depth Label Sequences of Block Tree Patterns in Acquisition of  
Characteristic Block Preserving Outerplanar Graph Patterns

徳原 史也<sup>\*1</sup> 宮原 哲浩<sup>\*1</sup> 鈴木 祐介<sup>\*1</sup> 内田 智之<sup>\*1</sup> 久保山 哲二<sup>\*2</sup>  
Fumiya Tokuhara Tetsuhiro Miyahara Yusuke Suzuki Tomoyuki Uchida Tetsuji Kuboyama

<sup>\*1</sup>広島市立大学情報科学研究科 Graduate School of Information Sciences, Hiroshima City University  
<sup>\*2</sup>学習院大学計算機センター Computer Centre, Gakushuin University

Machine learning and data mining from graph structured data are studied intensively. We use block preserving outerplanar graph patterns having structured variables for expressing structural features of outerplanar graphs. We consider depth label sequences of block tree patterns, which are tree representations of block preserving outerplanar graph patterns. Preliminary experimental results show that using the depth label sequences is effective in reducing the run time of our evolutionary learning method, which acquires characteristic block preserving outerplanar graph patterns from positive and negative outerplanar graph data.

## 1. はじめに

グラフ構造を持つデータに関する機械学習やデータマイニングが注目されている。例えば、多くの化合物は外平面的グラフの構造を持っており、化合物データからグラフ構造の特徴を獲得する研究が知られている [Horvath 10]。

遺伝的プログラミング (Genetic Programming, GP) [Koza 92, Banzhaf 98] とは、遺伝的アルゴリズムの遺伝子型を木構造のような構造的表現を扱えるように拡張した進化的手法である。外平面的グラフのグラフ構造の特徴を表現するブロック保存型外平面的グラフパターン (Block Preserving Outerplanar graph pattern, BPO グラフパターン) が提案されている [Sasaki 08, Yamasaki 09]。ブロック木パターンとは BPO グラフパターンの木表現である。正事例と負事例の外平面的グラフの集合から特徴的な BPO グラフパターンを獲得する遺伝的プログラミングによる手法 [Ouchiyama 15, 大内山 16, Tokuhara 16a] において、ブロック木パターンの正規形表現 [徳原 16] を利用して深さラベル列を定義し、実行時間を削減する方法を提案している [Tokuhara 16b]、本稿で実験結果を報告する。

関連研究として、BPO グラフパターンにより定義されるグラフ言語のクラスが正事例から多項式時間帰納推論可能であることが示され、正事例からの頻出 BPO グラフパターン列挙アルゴリズムが提案されている [Sasaki 08, Yamasaki 09]。正事例と負事例からなる木構造データから特徴的な木パターンを獲得する遺伝的プログラミングによる手法 [Nagamine 07, Nakai 13, Miyahara 14]、正事例と負事例からなるグラフ構造データから特徴的なグラフパターンを獲得する遺伝的プログラミングによる手法 [Nagai 12]、及びグラフ構造データに対する進化的手法 [Katagiri 00, Shirakawa 07] がある。

## 2. 準備

BPO グラフパターンとその木表現であるブロック木パターンについて説明する [Sasaki 08, Yamasaki 09, 大内山 16]。ラベル付きグラフとは、頂点と辺がラベル付けされたグラフをい

う。すべての頂点が外平面に接するように平面埋め込み可能であるグラフを外平面的グラフとよぶ。本稿では、連結なラベル付き外平面的グラフのみを扱うため、以後連結なラベル付き外平面的グラフを単に外平面的グラフということにする。連結グラフ  $G$  において、削除することで非連結にできる頂点と辺をそれぞれ  $G$  のカット点とブリッジという。本稿では、頂点数 3 以上のカット点をもたない極大な部分グラフ (2 重連結成分) を  $G$  のブロックという。外平面的グラフのいくつかのブリッジを 2 頂点からなる変数 (ブリッジ変数) で置き換え、1 頂点からなる変数 (末端変数) を頂点に付加して得られるグラフパターンを **BPO** グラフパターンとよぶ。BPO グラフパターンの例を図 1 に示す。外平面的グラフ  $G$  と BPO グラフパターン  $p$  に対し、 $p$  のすべての変数を適当な外平面的グラフで置き換えることによって  $G$  と同型な外平面グラフが得られるとき、 $p$  と  $G$  はマッチするという。図 1 において、BPO グラフパターン  $p$  と外平面的グラフ  $G$  はマッチする。なぜならば、図に付記した  $v_1 = u_1$  のような頂点の同一視により、 $p$  の変数  $x, y, z$  を、それぞれ外平面的グラフ  $g_1, g_2, g_3$  と置き換えることで  $G$  と同型な外平面グラフを得ることができるからである。

BPO グラフパターン  $p$  の各ブロック  $B$  を次の操作 (1)-(3) を適用して置き換えることにより得られる、根無し無順序木の構造を持つグラフパターンを  $p$  のブロック木パターン [Sasaki 08, Yamasaki 09] とよび、 $t(p)$  と表す。(1) ブロック  $B$  内の辺の接続情報をラベル (ブロックラベル) としてもつ新しい頂点 (ブロック頂点)  $v_B$  を追加する。(2)  $B$  中のすべての辺を削除する。(3)  $B$  中の各頂点と頂点  $v_B$  とを辺で結ぶ。図 1 にブロック木パターンの例を示す。

以下で述べる特徴的な BPO グラフパターン獲得問題を解くための、遺伝的プログラミングによる手法を提案している [Ouchiyama 15, 大内山 16, Tokuhara 16a]。

特徴的な BPO グラフパターン獲得問題

入力: 正事例、及び負事例からなる外平面的グラフの有限集合  $D$ 。

問題:  $D$  に関する適合度の高い BPO グラフパターンを獲得せよ。

ここで、BPO グラフパターン  $p$  の  $D$  に関する適合度  $fitness_D(p)$  を、 $fitness_D(p) = (D$  の正事例に  $p$  がマッチする割合  $+ D$  の負事例に  $p$  がマッチしない割合  $)/2$  と定義する。

連絡先: 徳原史也, 広島市立大学情報科学研究科, 〒731-3194, 広島市安佐南区大塚東 3-4-1, mb67015@e.hiroshima-cu.ac.jp

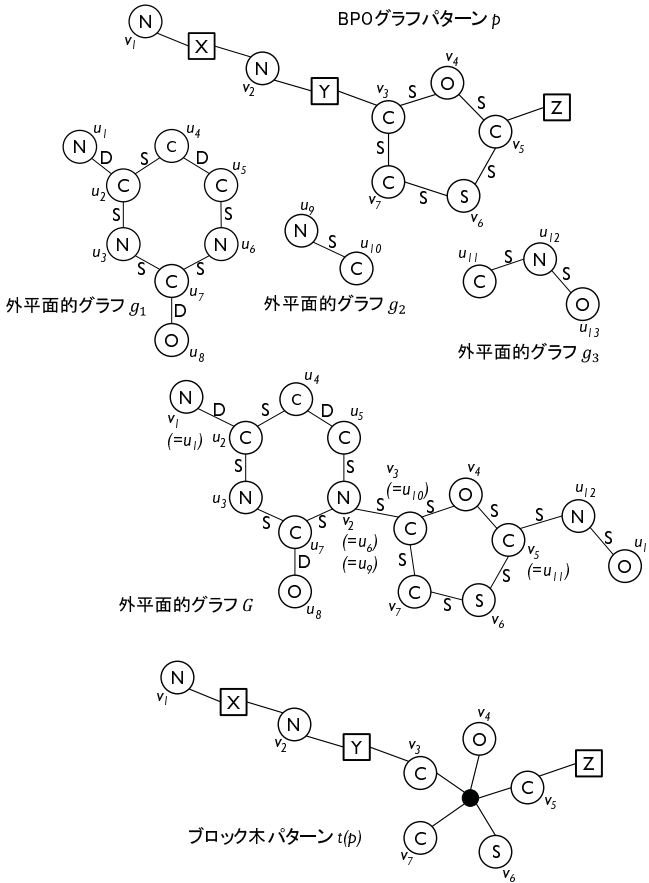


図 1: BPO グラフパターン  $p$ , 外平面的グラフ  $g_1, g_2, g_3, G$ , ブロック木パターン  $t(p)$ .

### 3. ブロック木パターンの深さラベル列

ブロック木パターンの深さラベル列の定義と、深さラベル列を計算するための手法を与える。

#### 3.1 根付きブロック木パターン

ブロック木パターンの深さラベル列を求めるために、まずブロック木パターンの根について説明する。自由木(根無し無順序木)の中心の決定方法を拡張してブロック木パターンの根を決定する。ブロック木パターンは、BPO グラフパターンの木表現であり、いくつかの内部頂点の子が順序を持っている特別な根無し無順序木である。根付きブロック木パターンとは、ブロック木パターンに根となる特別な頂点を指定したものである。ブロック木パターンの中心は、自由木の中心と同様に、自分以外の頂点までの距離の最大値が最小となる頂点のことである。ブロック木パターンの中心は1つまたは2つ存在する。ブロック木パターンの中心、または中心に隣接する頂点をブロック木パターンの根と指定する。根付きブロック木パターンでは、ブロック頂点の子はブロックのハミルトンサイクルの走査における時計回り順または反時計回り順と一致するような兄弟間の順序を持つ。根付きブロック木パターンのブロック頂点以外の頂点は、子の順序を持たない。

#### 3.2 根付きブロック木パターンの正規形表現

無順序木の正規形表現 [Nakano 03, Asai 03] と自由木の正規形表現 [Chi 03] を拡張し、根付きブロック木パターンの正

規形表現を定義する。

根付きブロック木パターン  $t$  に対し、 $V(t)$  を  $t$  の頂点集合とする。 $V(t) \times V(t)$  の部分集合  $>_t$  が次の条件 (1)-(3) をみたすとき、 $>_t$  は  $t$  の兄弟関係であるという。(1) 関係  $>_t$  は  $V(t)$  の半順序集合である。(2) 関係  $>_t$  は  $t$  のすべての内部頂点に対して兄弟関係を与える。(3) 関係  $>_t$  は、 $t$  の任意のブロック頂点に対し、ブロックラベルに基づく時計回り順または反時計回り順と一致するような、ブロック頂点の子の順序を与える。

兄弟関係が与えられた根付きブロック木パターンを順序ブロック木パターンとよぶ。順序ブロック木パターン  $T$  が根付きブロック木パターン  $t$  と兄弟関係  $>_t$  から作成されたならば、 $T$  のブロックラベルの情報は兄弟関係  $>_t$  に応じて書き直される。

根付きブロック木パターン  $t$  に、 $t$  上の兄弟関係  $>_t$  を与えることで得られる順序ブロック木パターンを、 $t$  の順序木表現とよぶ。根付きブロック木パターン  $t$  に対し、兄弟関係の与え方により  $t$  の順序木表現は複数存在する。根付きブロック木パターン  $t$  に対し、 $t$  の順序木表現のすべての集合を  $Rep(t)$  とする。このとき順序ブロック木パターン  $T_1, T_2$  に対して、ある根付きブロック木パターン  $t$  が存在し、 $T_1, T_2 \in Rep(t)$  であるなら  $T_1$  と  $T_2$  は同値であるという。

順序ブロック木パターンの深さラベル列を次のように定める。 $T$  を順序ブロック木パターンとする。 $\Lambda$  および  $\Delta$  をそれぞれ  $T$  の頂点ラベルと辺ラベルの集合とする。ここで、末端変数は頂点でないことに注意する。 $N = \{0, 1, 2, \dots\}$  とする。 $T$  中の頂点  $v$  に対し、根から  $v$  までの経路の長さ、すなわち根から  $v$  までの間の辺とブリッジ変数の数を  $dep(v) \in N$  と表す。頂点  $v$  の頂点ラベルを  $vLabel(v) \in \Lambda \cup B$  と表す。ここで  $B$  は、 $B \cap \Lambda = \emptyset$  を満たすブロックラベルの集合である。頂点  $v$  に対し、 $eLabel(v) \in \{\delta_{root}, \delta_{block}, \delta_{bridge}\} \cup \Delta$  を次の (1)-(4) のように定める。ここで  $\{\delta_{root}, \delta_{block}, \delta_{bridge}\} \cap \Delta = \emptyset$  を満たすとする。(1) 頂点  $v$  が根のとき  $eLabel(v) = \delta_{root}$ 。(2) 頂点  $v$  または  $v$  の親がブロック頂点のとき  $eLabel(v) = \delta_{block}$ 。(3) 頂点  $v$  とその親がブリッジ変数で隣接しているとき  $eLabel(v) = \delta_{bridge}$ 。(4) それ以外のとき  $eLabel(v)$  は頂点  $v$  とその親の間の辺ラベルとする。頂点  $v$  に対し、 $v$  が末端変数と接続しているかどうかを  $term(v) \in \{0, 1\}$  で表す。 $v$  が末端変数と接続しているとき、 $term(v) = 1$  とする。 $T$  の頂点  $v_i$  に対し、4 組  $(dep(v_i), term(v_i), vLabel(v_i), eLabel(v_i)) \in (N \times \{0, 1\} \times (\Lambda \cup B) \times (\{\delta_{root}, \delta_{block}, \delta_{bridge}\} \cup \Delta))$  を頂点  $v_i$  の深さラベル組とよぶ。

頂点数  $k$  の順序ブロック木パターン  $T$  に対し、 $T$  の深さラベル列は、 $T$  の各頂点の深さラベル組を頂点の行きがけ順 (priororder) に並べたものであり、 $((dep(v_1), term(v_1), vLabel(v_1), eLabel(v_1)), \dots, (dep(v_k), term(v_k), vLabel(v_k), eLabel(v_k))) \in (N \times \{0, 1\} \times (\Lambda \cup B) \times (\{\delta_{root}, \delta_{block}, \delta_{bridge}\} \cup \Delta))^k$  と表される。ここで  $(v_1, \dots, v_k)$  は  $T$  の頂点を行きがけ順に並べたものである。 $T$  の深さラベル列を  $DLS(T)$  で表す。

BPO グラフパターン  $p$  のブロック木パターン  $t(p)$  に対し、根  $r$  を指定した根付きブロック木パターンを  $t(p, r)$  と表す。 $t(p, r)$  に兄弟関係が与えられた時に、深さラベル列を計算する手続 CALCDLS を [Tokuhara 16b] で示した。図 2 に順序ブロック木パターンの深さラベル列の例を示す。

次に深さラベル組上の全順序  $\geq_{dlt} \subseteq (N \times \{0, 1\} \times (\Lambda \cup B) \times (\{\delta_{root}, \delta_{block}, \delta_{bridge}\} \cup \Delta))^2$  を定義する。 $\Lambda \cup B$  上の順序  $>_{vl}$  を、2つのブロックラベルに対しては  $B$  上の順序、2つの頂点ラベルに対しては  $\Lambda$  上の順序、任意のブロックラベ

ル  $b$  と頂点ラベル  $\lambda \in \Lambda$  に対しては  $b >_{vl} \lambda$  と定義する. 同様に,  $\{\delta_{root}, \delta_{block}, \delta_{bridge}\} \cup \Delta$  上の順序  $>_{el}$  を, 2つの辺ラベルに対しては  $\Delta$  上の順序, 任意の辺ラベル  $\delta \in \Delta$  と他のラベルに対しては  $\delta_{root} >_{el} \delta_{block} >_{el} \delta_{bridge} >_{el} \delta$  と定義する. 任意の深さラベル組  $(d_1, t_1, vl_1, el_1)$  と  $(d_2, t_2, vl_2, el_2) \in (N \times \{0, 1\} \times (\Lambda \cup B) \times (\{\delta_{root}, \delta_{block}, \delta_{bridge}\} \cup \Delta))$  に対し, 関係  $(d_1, t_1, vl_1, el_1) \geq_{dt} (d_2, t_2, vl_2, el_2)$  であるのは,

- (1)  $d_1 > d_2$ ,
- (2)  $d_1 = d_2$  and  $t_1 > t_2$ ,
- (3)  $d_1 = d_2$  and  $t_1 = t_2$  and  $vl_1 >_{vl} vl_2$ ,
- (4)  $d_1 = d_2$  and  $t_1 = t_2$  and  $vl_1 = vl_2$  and  $el_1 >_{el} el_2$ ,
- (5)  $d_1 = d_2$  and  $t_1 = t_2$  and  $vl_1 = vl_2$  and  $el_1 = el_2$ .

のいずれかのときである.

最後に, 深さラベル組上の全順序  $\geq_{dt}$  から生成される, 深さラベル列上の辞書式順序  $\geq_{lex}$  を, 深さラベル列上の全順序として定義する. 順序ブロック木パターン  $T$  が,  $T$  と同値な任意の  $S$  に対して,  $DLS(T) \geq_{lex} DLS(S)$  を満たすとき, 根付き正規形パターンとよぶ.

根付き正規形パターン  $T$  が根付きブロック木パターン  $t$  の順序木表現であるならば,  $T$  の深さラベル列  $DLS(T)$  を  $t$  の根付き正規形表現とよぶ. 図 2 に, 根付き正規形表現の例を示す. 順序ブロック木パターン  $T_1, T_2, T_3$  はブロック木パターン  $t(p)$  から得られる.  $T_1, T_2, T_3$  は同値な順序ブロック木パターンである. このとき,  $DLS(T_1) \geq_{lex} DLS(T_2) \geq_{lex} DLS(T_3)$  であるので,  $T_2, T_3$  は根付き正規形パターンでないことが分かる. また  $T_1$  は同値な任意の順序木パターン  $S$  に対して,  $DLS(T_1) \geq_{lex} DLS(S)$  を満たすので根付き正規形パターンである.

### 3.3 ブロック木パターンの深さラベル列

ブロック木パターンの深さラベル列を定義する. BPO グラフパターン  $p$  のブロック木パターン  $t(p)$  に対し,  $t(p)$  の正規形表現を, 根付きブロック木パターン  $t(p, r)$  の根付き正規形表現と定義する. ここで根  $r$  は次の条件 1-4 のいずれかを満たすものとする. (条件 1)  $t(p)$  はただ 1つの中心  $r$  を持ち,  $r$  がブロック頂点でない. (条件 2)  $t(p)$  はただ 1つの中心  $c$  を持ち,  $c$  はブロック頂点である.  $r$  は  $c$  に隣接した頂点であり,  $t(p, r)$  の根付き正規形表現は,  $c$  の隣接頂点を根とした根付きブロック木パターンの根付き正規形表現の中で辞書式順序  $>_{lex}$  で最も大きい. (条件 3)  $t(p)$  は 2つの中心  $r, r'$  を持ち,  $r$  と  $r'$  はともにブロック頂点ではない.  $t(p, r)$  の根付き正規形表現は,  $t(p, r')$  の根付き正規形表現よりも辞書式順序  $>_{lex}$  において大きい. (条件 4)  $t(p)$  は 2つの中心  $r, r'$  を持ち,  $r$  はブロック頂点でなく  $r'$  はブロック頂点である.

ブロック木パターンの正規形表現を計算するアルゴリズム CALCCANONICAL を [Tokuhara 16b] で示した. BPO グラフパターン  $p$  のブロック木パターン  $t(p)$  の正規形表現を, ブロック木パターンの深さラベル列とよぶ. 図 3 にブロック木パターンとその深さラベル列の例を示す.

## 4. 実験

ブロック木パターンの深さラベル列を計算するアルゴリズムを Intel Xeon CPU E5-2630 v2 2.60GHz プロセッサ, 実装メモリ 32.0GB の Windows10 Pro 64bit OS 上に Java 言語で実装した. 我々の従来手法である, 正事例と負事例の外平面的グラフの集合から特徴的な BPO グラフパターンを獲得する遺伝的プログラミングによる手法 [Ouchiyama 15, 大内山 16, Tokuhara 16a] においては, 深さラベル列を利用していない.

提案手法では, 正事例と負事例の外平面的グラフの集合から特徴的な BPO グラフパターンを獲得する遺伝的プログラミングによる従来手法を以下のように拡張した. 個体であるブロック木パターンの深さラベル列をキーとし, 個体の適合度を値として HashMap 表で管理し, 個体の適合度を求める際に既に HashMap 表にキーとして存在しないものについてだけ, BPO グラフパターンマッチングアルゴリズム [Sasaki 08, Yamasaki 09] を用いてデータとマッチするかを計算して適合度を計算する. この HashMap 表を用いて, 各世代の集団の同型でない BPO グラフパターンの個数を求めて, 集団の多様性を容易に評価することができる.

実験データとして, NCI データベース [NCI 99] のファイル “CAD2DA99.sdz” から CA(Confirmed Active) とラベル付けされた外平面的グラフとみなせるデータ 87 個を正事例とし, CI(Confirmed Inactive) とラベル付けされた外平面的グラフとみなせるデータから 87 個を選んで負事例として使用した. 提案手法と従来手法で, それぞれ 10 試行の特徴的な BPO グラフパターン獲得を行った.

提案手法と従来手法に共通する GP のパラメータは以下のとおりである. 集団サイズ (集団の個体数): 50, 複製確率: 0.05, 交叉確率: 0.50, 突然変異確率: 0.45 (部分パターン交換: 0.30, 部分パターン追加: 0.30, 部分パターン削除: 0.30, 頂点ラベル交換: 0.05, 辺ラベル交換: 0.05), 選択方法: ルーレット選択, トーナメント選択 (サイズ 2), エリート保存 (サイズ 3), 最大世代数: 200.

提案手法と従来手法における実行時間の 10 試行の平均は, それぞれ 294 秒と 396 秒である. 最終世代の最良個体の適合度の 10 試行の平均は, 0.755 である. ブロック木パターンの深さラベル列を利用する提案手法では, これを利用しない従来手法に比べて実行時間を削減できることがわかる. 各世代の集団における同型でない BPO グラフパターン (個体) の個数の 10 試行の平均を図 4 に示す.

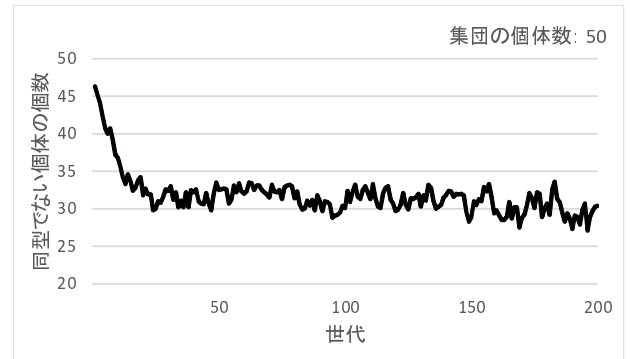


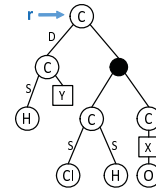
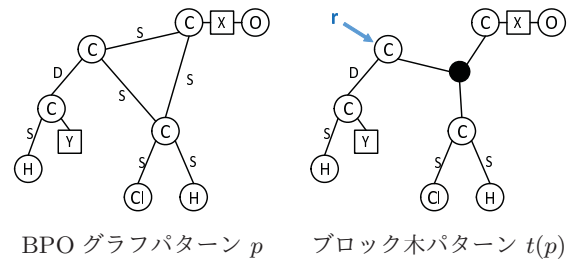
図 4: 同型でない BPO グラフパターン (個体) の個数

## 5. おわりに

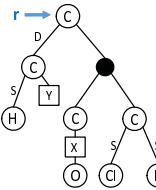
本研究では, 正事例と負事例の外平面的グラフから遺伝的プログラミングによる特徴的な BPO グラフパターンを獲得する手法において, ブロック木パターンの深さラベル列を利用する方法を提案した. 深さラベル列の利用が実行時間の削減, 集団の多様性の評価に有用であることがわかった. 提案手法を大規模なグラフ構造データに適用して有効性を確認することなどが今後の課題である.

## 参考文献

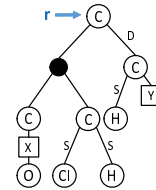
- [Asai 03] T.Asai et al.: Discovery of Frequent Substructures in Large Unordered Trees, *Proc. DS-2003, Springer-Verlag, LNAI 2843*, pp. 47–61 (2003)
- [Banzhaf 98] W.Banzhaf et al.: *Genetic Programming: An Introduction : On the Automatic Evolution of Computer Programs and Its Applications*, Morgan Kaufmann (1998)
- [Chi 03] Y.Chi et al.: Mining Frequent Rooted Trees and Free Trees using Canonical Forms, *UCLA Computer Science Department Technical Report CSD-TR No.030043* (2003)
- [Horvath 10] T.Horvath et al.: Frequent Subgraph Mining in Outerplanar Graphs, *Data Mining and Knowledge Discovery*, Vol. 21, pp. 472–508 (2010)
- [NCI 99] National Cancer Institute: The NCI Open Database, *Release 1 Files* (1999)
- [Katagiri 00] H.Katagiri et al.: Genetic Network Programming - Application to Intelligent Agents, *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, pp. 3829–3834 (2000)
- [Koza 92] J.R.Koza: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press (1992)
- [Miyahara 14] T.Miyahara and T.Kuboyama: Learning of Glycan Motifs Using Genetic Programming and Various Fitness Functions, *Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, Vol. vol.18(3), pp. 401–408 (2014)
- [Nagai 12] S.Nagai et al.: Acquisition of Characteristic TTSP Graph Patterns by Genetic Programming, *Proc. IIAI AAI 2012*, pp. 340–344 (2012)
- [Nagamine 07] M.Nagamine et al.: A Genetic Programming Approach to Extraction of Glycan Motifs using Tree Structured Patterns, *Proc. AI-2007, Springer-Verlag, Vol. 4830*, pp. 150–159 (2007)
- [Nakai 13] S.Nakai et al.: Acquisition of Characteristic Tree Patterns with VLDC's by Genetic Programming and Edit Distance, *Proc. IIAI AAI 2013*, pp. 147–151 (2013)
- [Nakano 03] S.Nakano and T.Uno: Efficient generation of rooted trees, *NII Technical Report, NII-2003-005E, National Institute of Informatics, Japan* (2003)
- [Ouchiyama 15] Y.Ouchiyama et al.: Acquisition of Characteristic Block Preserving Outerplanar Graph Patterns from Positive and Negative Data using Genetic Programming and Tree Representation of Graph Patterns, *Proc. IWCIA 2015*, pp. 95–101 (2015)
- [Sasaki 08] Y.Sasaki et al.: Mining of Frequent Block Preserving Outerplanar Graph Structured Patterns, *Proc. ILP 2007, Springer-Verlag LNAI 4894*, pp. 239–253 (2008)
- [Shirakawa 07] S.Shirakawa et al.: Graph Structured Program Evolution, *Proc. GECCO 2007*, pp. 1686–1693 (2007)
- [Tokuhara 16a] F.Tokuhara et al.: Acquisition of Characteristic Block Preserving Outerplanar Graph Patterns by Genetic Programming using Label Information, *Proc. IIAI AAI 2016*, pp. 203–210 (2016)
- [Tokuhara 16b] F.Tokuhara et al.: Using canonical representations of block tree patterns in acquisition of characteristic block preserving outerplanar graph patterns, *Proc. IWCIA 2016*, pp. 93–99 (2016)
- [Yamasaki 09] H.Yamasaki et al.: Learning block-preserving graph patterns and its application to data mining, *Machine Learning*, Vol. 76, pp. 137–173 (2009)
- [大内山 16] 大内山 祐斗 ほか: グラフパターンの木表現を用いた遺伝的プログラミングによる特徴的なブロック保存型外平面的グラフパターンの獲得, 2016 年度人工知能学会全国大会論文集, 3I4–4 (2016)
- [徳原 16] 徳原 史也 ほか: 特徴的なブロック保存型外平面的グラフパターンの獲得におけるブロック木パターンの正規形表現の利用, 2016 IEEE SMC Hiroshima Chapter 若手研究会講演論文集, pp. 138–143 (2016)



$$DLS(T_1) = ((0,0,C,\#R),(1,1,C,D),(2,0,H,S),(1,0,((1-2,S)(2-3,S)(3-1,S)),\#B),(2,0,C,\#B),(3,0,Cl,S),(3,0,H,S),(2,0,C,\#B),(3,0,O,\#V))$$

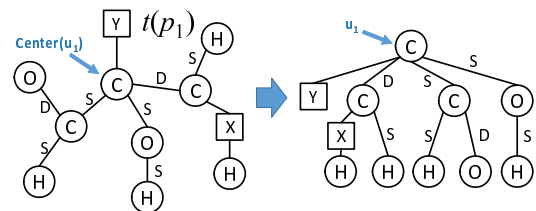


$$DLS(T_2) = ((0,0,C,\#R),(1,1,C,D),(2,0,H,S),(1,0,((1-2,S)(2-3,S)(3-1,S)),\#B),(2,0,C,\#B),(3,0,O,\#V),(2,0,C,\#B),(3,0,Cl,S),(3,0,H,S))$$



$$DLS(T_3) = ((0,0,C,\#R),(1,0,((1-2,S)(2-3,S)(3-1,S)),\#B),(2,0,C,\#B),(3,0,O,\#V),(2,0,C,\#B),(3,0,Cl,S),(3,0,H,S),(1,1,C,D),(2,0,H,S))$$

図 2: BPO グラフパターン  $p$  とそのブロック木パターン  $t(p)$ , および  $t(p, r)$  から得られる順序ブロック木パターン  $T_1, T_2, T_3$ . 上記の深さラベル列では  $\#R, \#B, \#V$  は  $\delta_{root}, \delta_{block}, \delta_{bridge}$  をそれぞれ表す.  $((1-2,S)(2-3,S)(3-1,S))$  はブロックラベルである, ここで  $C >_{vl} Cl >_{vl} H >_{vl} O, D >_{el} S$  とする.



$$DLS(t(p_1, u_1)) = ((0,1,C,\#R),(1,0,C,D),(2,0,H,\#V),(2,0,H,S),(1,0,C,S),(2,0,H,S),(2,0,O,D),(1,0,O,S),(2,0,H,S))$$

図 3: ブロック木パターン  $t(p_1)$  とその深さラベル列  $DLS(t(p_1, u_1))$ . 図中の根  $u_1$  は (条件 1) を満たす. 上記の深さラベル列では  $\#R, \#V$  は  $\delta_{root}, \delta_{bridge}$  をそれぞれ表す.  $C >_{vl} Cl >_{vl} H >_{vl} N >_{vl} O, D >_{el} S$  とする.