情報処理学会論文誌　データベース　Vol. 1　No. 1　26–35 (June 2008)

構造データ集合からなるグラフデータベースからの頻出パターン発見

山本　翼[†1]　尾崎知伸[†2]　大川剛直[†1]

本稿では，生物情報学における代謝パスウェイや社会ネットワークなどに対するより詳細な分析手段を提供することを目的に，各頂点にアイテム集合や系列などの構造データの集合を持つ複雑なグラフデータベース，すなわち複合構造グラフデータベースを対象とした頻出パターン発見手法を提案する．また，得られるパターン数の増大という頻出パターン発見における問題に対処するため，パターン中の各頂点を，利用者による制約を満たしかつ代表的なものに限定する枝刈り手法を導入する．実データを用いた実験により，既存研究では発見できなかったパターンを発見できることを確認した．

Discovery of Frequent Patterns in Multi-structured Graph Databases

Tsubasa Yamamoto,[†1] Tomonobu Ozaki[†2] and Takenao Ohkawa[†1]

In this paper, as one of the tools for precise analysis of complex networks such as metabolic pathways in bioinformatics and social networks, we propose an algorithm for mining frequent patterns in multi-structured graph databases in which each vertex consists of a set of structured data such as item sets and sequences. Furthermore, we also propose two pruning mechanisms to exclude uninteresting patterns to alleviate the problem that huge number of patterns will be discovered. The effectiveness of the proposed algorithms is confirmed through the experiments with two real datasets. In addition, the proposed algorithms succeeded in finding some patterns which were not discovered by conventional graph miners.

1. はじめに

グラフは，タンパク質やLSI，ウェブのハイパーリンク，XMLといった構造データを表現する際によく利用される．グラフデータベース中に頻出する部分グラフの発見は，グラフマイニングにおいて重要な問題の1つであり[4),18)]，近年さかんに研究されている[1),7),8),12),19),20)]．一方，より複雑なグラフデータベースの例として，代謝パスウェイのデータベースKEGG[⋆1]を考える．代謝は，生物の化学反応の総体であり，化学反応は酵素によって引き起こされ，ある化合物を別の化合物へと変換する．代謝パスウェイは，この化学反応の大きなネットワークであり，グラフ構造を持つデータと見なすことができる．代謝パスウェイの各頂点は，酵素や化合物，遺伝子などに対応する．酵素や化合物は，それらを識別する名前以外に，アミノ酸配列や，構造式，疎水性といった様々な情報を含んでいる．したがって，代謝パスウェイをより自然に表現するためには，単なるグラフでは不十分であり，各頂点にアミノ酸配列や構造式などを持つグラフが必要とされる．本稿では，アイテム集合や系列などの構造データの集合を頂点とする特殊なグラフを**複合構造グラフ**と呼ぶ．既存の頻出部分グラフ発見手法の多くは，グラフの各頂点を1つのアトムとして扱うので，頂点が持つ部分的なパターンまでを考慮した部分グラフを発見することができない．その一方で，複合構造グラフは今後ますますの増大が予想され，それらを扱うことのできる柔軟な手法や枠組みを構築することは重要な課題であると考えられる．

これらのことを背景に，本稿では，複合構造グラフデータベースからの頻出パターン発見アルゴリズム **FMG** を提案する．FMGは，グラフ構造列挙手法を用いた(1) **外部構造**（グラフ構造）の列挙と，集合や系列列挙手法などを用いた(2) **内部構造**（頂点構造）の列挙を組み合わせることで，複合構造データベースにおける頻出パターンの完全な列挙を実現する．さらに本稿では，得られるパターン数の増大という頻出パターン発見の欠点に対処するため，パターン内の頂点を，(1) 利用者による制約を満たす，(2) 代表的なパターンに限定する手法 **CCFMG** を提案する．

以下に本稿の構成を示す．2章では，準備として，用語を導入するとともに，FMGの基

†1 神戸大学大学院工学研究科
Graduate School of Engineering, Kobe University
†2 神戸大学自然科学系先端融合研究環
Organization of Advanced Science and Technology, Kobe University
⋆1 http://www.genome.ad.jp/kegg/

となるグラフパターンの列挙について述べる．ついで 3 章で FMG，4 章で CCFMG をそれぞれ提案する．5 章で関連研究について述べ，6 章で実験結果を示す．最後に 7 章でまとめを行う．

2. 準　　備

2.1 用語の導入と問題の形式化

本節では，文献 1)，20) などに従い，定義といくつかの記法を導入するとともに，本稿で対象とする問題の形式的な定義を与える．

複合構造グラフ $G = (V_G, E_G)$ は，頂点集合 V_G と辺集合 E_G から構成される．各頂点 $v \in V_G$ は，ちょうど n 個のデータからなる集合を持つ．頂点 v が持つ集合を $\mathrm{s}(v) = [elm_1^v, \cdots, elm_n^v] \in [\mathrm{dom}(A_1), \cdots, \mathrm{dom}(A_n)]$ と表記する．ここで $\mathrm{dom}(A_i)$ は，i 番目の属性 A_i の領域，すなわち集合や系列などの属性のクラスを表す．また，頂点 v と v' を結ぶ辺を (v, v') と表記する．**図 1** に複合構造グラフの例を示す．各頂点はちょうど 2 つのデータを持つ．また，第 1 属性のクラスをアイテム集合，第 2 属性のクラスを系列と仮定する．$v_{13} \in V_{G_1}$ に対し，$\mathrm{s}(v_{13}) = [\{a, b, c\}, \langle AACC \rangle]$ である．なお ϕ は，空集合または空列を表す．

属性 A_i におけるパターン p，$q \in \mathrm{dom}(A_i)$ に対し，p が q に含まれるとき，p は q より一般的なパターンであると定義し，$p \preceq q$ と表記する．$\mathrm{s}(v) = [elm_1^v, \cdots, elm_n^v]$ なる頂点 v と，パターン列 $lp = [p_1, \cdots, p_n] \in [\mathrm{dom}(A_1), \cdots, \mathrm{dom}(A_n)]$ に対し，すべての p_i が対応する要素

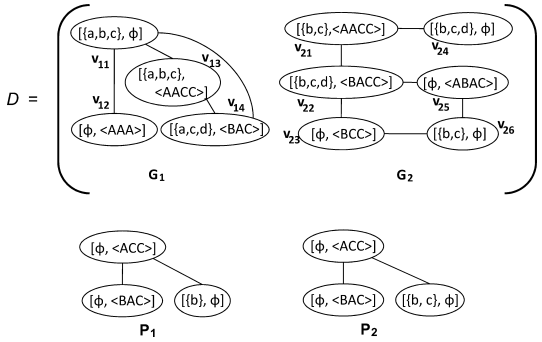


図 1　複合構造グラフデータベースとパターンの例
Fig. 1　An example of multi-structured graph database and patterns.

elm_i^v を被覆するとき，すなわち $\forall i\,(p_i \preceq elm_i^v)$ が成り立つとき，**lp は頂点 v を被覆する**と定義し，$lp \preceq v$ と表記する．2 つの複合構造グラフ $G = (V_G, E_G)$，$G' = (V_{G'}, E_{G'})$ に対し，(1) $\forall (u, v) \in E_G\,((f(u), f(v)) \in E_{G'})$，(2) $\forall v \in V_G\,(v \preceq f(v))$ を満たす単射 $f : V_G \to V_{G'}$ が存在するとき，G を G' の**部分グラフ**，G' を G の**上位グラフ**と呼び，$G \subseteq G'$ と表記する．単射 f により得られる G' 中の辺集合 $\{(f(u), f(v)) \in E_{G'} \mid (u, v) \in E_G\}$ を **G' における G の出現**と定義する．また，G' における G の出現の全体集合を $\Phi_{G'}^{G}$ と表記する．たとえば図 1 において，$\Phi_{G_1}^{P_1} = \{\{(v_{13}, v_{14}), (v_{13}, v_{11})\}\}$，$\Phi_{G_2}^{P_2} = \{\{(v_{21}, v_{22}), (v_{21}, v_{24})\}, \{(v_{22}, v_{25}), (v_{22}, v_{21})\}\}$ である．

複合構造グラフデータベース $D = \{G_1, G_2, \cdots, G_M\}$ に対し，パターン P の**支持度**を次のように定義する．

$$sup_D(P) = \frac{1}{M} \sum_{G \in D} O_P(G)$$
$$\text{where} \qquad O_P(G) = \begin{cases} 1 & (P \subseteq G) \\ 0 & (\text{otherwise}) \end{cases}$$

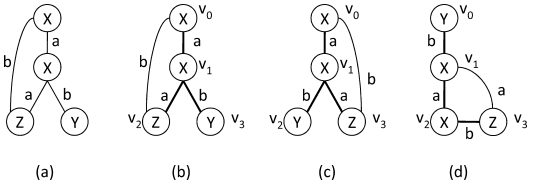
利用者により与えられる最小支持度 σ（$0 < \sigma \leq 1$）に対し，$sup_D(P) \geq \sigma$ を満たすパターンを**頻出パターン**と呼ぶ．本稿では，D と σ が与えられたとき，D 中のすべての頻出パターンを発見する問題について議論する．

2.2 最右拡張と標準形判定を用いた頻出部分グラフの発見

本節では，FMG の出発点として，ラベル付きグラフを対象とした最右拡張と標準形判定を用いた頻出部分グラフ発見アルゴリズム gSpan [20] を概観する．

グラフを深さ優先で探索し，全域木を構成することを考える．探索の開始点を根，最終点を**最右頂点**と呼ぶ．また，全域木上で根から最右頂点に至る経路を**最右パス**と呼ぶ．グラフの辺は全域木に含まれるかによって 2 種類に分けることができ，全域木を構成する辺を**前方辺**，それ以外の辺を**後方辺**と呼ぶ．**図 2** 中の (b)–(d) の各グラフは，(a) のグラフに対して深さ優先探索を行い，前方辺を太い実線で表したものである．また各グラフに対して，訪れた頂点順に順番 v_0，v_1，$\cdots$ を付与している．このとき，根は v_0，最右頂点は v_3 となる．(b) のグラフの最右パスは v_0–v_1–v_3 であり，(d) のグラフの最右パスは v_0–v_1–v_2–v_3 である．

gSpan は，グラフパターンに対し，(1) 最右パス上の頂点と最右頂点間に後方辺を追加する，もしくは (2) 最右パス上の頂点から新たな頂点をともなう前方辺を追加することで，新たなグラフパターンを得る．この操作を**最右拡張**と呼ぶ．また，グラフパターン g に辺 e を



code(b)= X (1 0 a X) (2 -1 a Z) (2 0 b X) (3 -1 b Y)
code(c)= X (1 0 a X) (2 -1 b Y) (3 -1 a Z) (3 0 b X)
code(d)= Y (1 0 b X) (2 -1 a X) (3 -2 b Z) (3 -1 a X)

図 2　ラベル付きグラフと全域木およびコードワード（文献 5) より引用）
Fig. 2　A labeled graph, spanning trees and codewords.

追加して得られる新たなパターンを $g \cdot e$ と表記する．

頂点のみからなるグラフパターンに対して最右拡張を繰り返し適用することで，すべてのグラフパターンを漏れなく列挙することが可能である[20)]．しかしその一方で，最右拡張だけでは，同型なグラフパターンが重複して生成されることとなる．gSpan では，グラフを文字列で表現し，同型なグラフパターン間で最小の文字列を持つグラフパターンのみを代表元として列挙することにより，この問題を回避している．ここで，グラフパターンの文字列表現をグラフパターンの**コードワード**，同型グラフパターンの代表元を**標準形**と呼ぶ．頂点 v に対し，最右拡張の繰返し適用により m 本の辺を追加したグラフパターン $g = v \cdot e_1 \cdot \ldots \cdot e_m$ に対するコードワードは，

$$code(g) = l\ (i_d^1\ -i_s^1\ a^1\ l_d^1) \cdots (i_d^m\ -i_s^m\ a^m\ l_d^m)$$

と定義される[1),20)]．ここで l は頂点 v のラベルを表す．また，i_d^j，i_s^j は辺 e_j における頂点番号，すなわち全域木を構成する際に訪れた順 $(i_s^j < i_d^j)$ を，a^j は，辺 e_j の辺ラベルを表す．l^j は，前方辺では i_d^j に対応する頂点のラベル，後方辺では i_s^j に対応する頂点のラベルである．図 2 において，数値の小さい順およびアルファベット順を前提とすると，各コードワードの大小関係は $code(b) < code(c) < code(d)$ となる．また同型なグラフパターンにおいて (b) が最小のコードワードを持つので，(b) が標準形となる．

図 3 に，最右拡張と標準形判定を用いた深さ優先探索による頻出部分グラフ発見アルゴリズム gSpan を示す．図中において，$\mathcal{L}$ は頂点ラベルの全集合を，isCanonical(g) はグラフパターン g が標準形であるときに真となる関数をそれぞれ表す．また，gSpan-Enum 中の 1 行目が標準形判定に，3–4 行目が最右拡張に対応する．

Algorithm gSpan(D, σ)
1: **for each** $v \in \mathcal{L}$
2:　**if** $sup_D(v) \geq \sigma$ **then**
3:　　gSpan-Enum(v, D, σ)
4: **end for**

Subroutine gSpan-Enum(s, D, σ)
1: **if** ¬isCanonical(s) **then** return
2: **output** s
3: scan D and set E be a set of all edges e
4:　s.t. s can be right most extended
5: **for each** $e \in E$
6:　**if** $sup_D(s \cdot e) \geq \sigma$ **then**
7:　　gSpan-Enum($s \cdot e$, D, σ)
8: **end for**

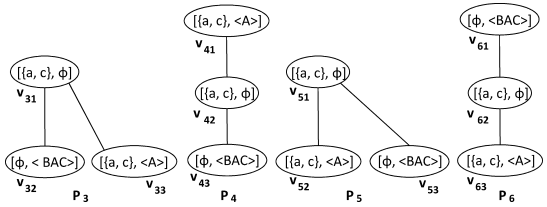
図 3　頻出部分グラフ発見アルゴリズム gSpan
Fig. 3　Pseudo code of gSpan.

3. 頻出複合構造グラフパターンの発見

本章では，複合構造グラフデータベースを対象とした，完全な頻出複合構造グラフパターン発見手法 FMG を提案する．

3.1 FMG の概要

FMG は，gSpan 同様，拡張による特殊化を用いた深さ優先探索に基づくパターン発見手法である．最初に列挙されるパターンは，大きさ 1 の構造データをちょうど 1 つ持ち，それ以外は ϕ の頂点だけからなるパターン I である．なお，構造データの大きさは，アイテム集合ならばアイテムの数，系列ならば系列の長さなど，各構造によってそれぞれ定義されるとする．たとえば，図 1 中のデータベース D を対象とした場合，パターン I としては，$[\{a\}, \phi]$，$[\{b\}, \phi]$，$[\{c\}, \phi]$，$[\{d\}, \phi]$，$[\phi, \langle A \rangle]$，$[\phi, \langle B \rangle]$，$[\phi, \langle C \rangle]$ が考えられる．FMG は，I に対して，(1) 最右頂点中の構造データ集合に対する拡張である**内部拡張**，および (2) グラフ構造に対する拡張である**外部拡張**の 2 種類の拡張を繰り返し適用することで，すべてのパターンを列挙する．以下，複合構造グラフにおける標準形と，内部・外部拡張について



code(P_3)= [{a,c},ϕ] (2 -1 – [ϕ,$\langle BAC\rangle$]) (3 -1 – [{a,c},$\langle A\rangle$])
code(P_4)= [{a,c},$\langle A\rangle$] (2 -1 – [{a,c},ϕ]) (3 -2 – [ϕ,$\langle BAC\rangle$])
code(P_5)= [{a,c},ϕ] (2 -1 – [{a,c},$\langle A\rangle$]) (3 -1 – [ϕ,$\langle BAC\rangle$])
code(P_6)= [ϕ,$\langle BAC\rangle$] (2 -1 – [{a,c},ϕ]) (3 -2 – [{a,c},$\langle A\rangle$])

図 4　同型の複合構造グラフパターンとコードワード
Fig. 4　Isomorphic multi-structured graph patterns and those codeword.

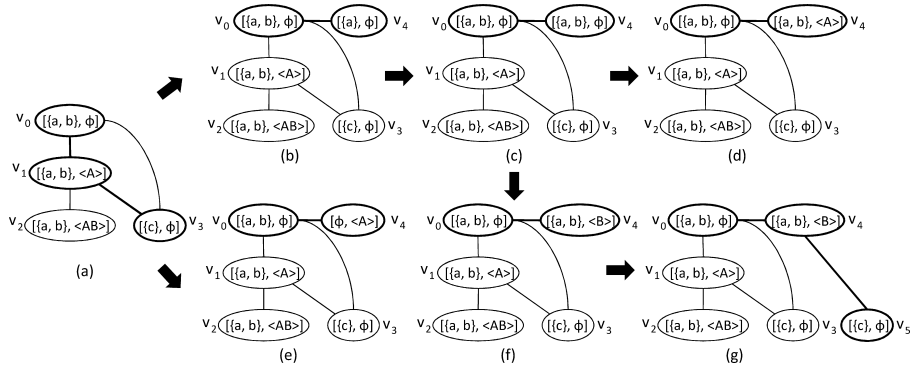


図 5　FMG の探索空間の一部
Fig. 5　A part of search space in FMG.

説明する．

3.2　複合構造グラフパターンの標準形

頂点 I に対し，m 本の辺を追加した複合構造グラフパターン $P = I \cdot e_1 \cdot \ldots \cdot e_m$ に対するコードワードを以下のように定義する．また，ラベル付きグラフの場合と同様，同型なパターン間において辞書式順で最小のコードワードを持つパターンを標準形と定義する．

$$code(P) = l(\mathrm{s}(I))\ (i_d^1\ -i_s^1\ a^1\ l(\mathrm{s}(I^1)))\cdots (i_d^m\ -i_s^m\ a^m\ l(\mathrm{s}(I^m)))$$

ここで，$l(\mathrm{s}(v))$ は頂点が持つデータ集合 $\mathrm{s}(v)$ に対する文字列表現である．定義から分かるように，複合構造グラフパターンのコードワードは，ラベル付きグラフにおける頂点ラベルを $l(\mathrm{s}(v))$ に置き換えたものとなっている．また，ϕ は，辞書式順で最小と定義する．**図 4** に，4 つの同型な複合構造グラフパターンとそのコードワードを示す．なお，すべての辺は，辺ラベル '–' を持つと仮定する．このとき，最小のコードワードは $code(P_6)$ であり，P_6 が標準形となる．

3.3　外部構造と内部構造の拡張

FMG は，gSpan 同様，最右パス上の頂点に対して辺を加えることにより，パターンを列挙する．また前方辺を追加する場合は，辺とともに，大きさ 1 の構造データを 1 つ持つ新たな頂点を追加する．**図 5** に，外部構造の拡張の例を示す．図中 (a) において，太い実線が最右パスを表す．ここで，v_0 に前方辺を加えることで，(b) や (e) のパターンが得られる．

gSpan では，頂点ラベルの集合が既知であり，追加する頂点の集合をあらかじめ決定することが可能である．加えて，標準形とならないグラフパターンを生成する頂点の追加をあらかじめ避けることが可能である．これに対し FMG は，頂点内の部分構造に着目するため，頂点を構成する（構造）パターン集合を列挙する必要がある．このため，前方辺によって追加された最右頂点の内部構造は，たとえばアイテム集合であれば文献 17) などのアイテム集合列挙アルゴリズム，系列であれば文献 14) などの系列列挙アルゴリズムといった，各属性のクラスに適したアルゴリズムを用い，より特殊な構造データへと拡張される．また，特定の属性のみを拡張するだけではなく，構造データ集合に対して別の属性のデータを追加することによって，ϕ でないデータ数の 1 つ多い構造データ集合も列挙する必要がある．この際，属性間に順序関係を定め，順序関係に従って構造データ集合の拡張を行うことで，重複を防ぐ．図 5 に，内部拡張の例を示す．(b) の最右頂点 v_4 の第 1 属性であるアイテム集合 $\{a\}$ を拡張することによって，(c) が得られる．また，(c) の第 2 属性に系列データを新たに追加することで，最右頂点のデータ数が 1 つ多い (d) や (f) が得られる．

ところで，現時点では標準形ではないパターンであっても，内部拡張により標準形が生成される可能性があるので，標準形となるまで繰り返し内部拡張を適用する必要がある（図 5 の (b) から (f)）．一方，最右頂点以外に起因して標準形とはなりえないパターンは，その後の拡張においても標準形とはなりえない．したがって，標準形を満たさないパターンに対しての，外部拡張の適用は行わない．

Algorithm FMG(D, σ)

1: set I be a set of all initial patterns
2: **for each** $p \in I$
3:　**if** $\sup_D(p) \geq \sigma$ **then**
4:　　FMG-Enum(p, D, σ)
5: **end for**

Subroutine FMG-Enum(p, D, σ)

1: set P_{in} be a set of patterns obtained
2:　by expanding internal structure of p
3: **for each** $s \in P_{in}$
4:　**if** $sup_D(s) \geq \sigma$ **then**
5:　　FMG-Enum(s, D, σ)
6: **end for**
7: **if** ¬isCanonical(p) **then** return
8: **output** p
9: scan D and set E be a set of all edges e
10:　s.t. p can be right most extended
11: **for each** $e \in E$
12:　**if** $sup_D(p \cdot e) \geq \sigma$ **then**
13:　　FMG-Enum($p \cdot e$, D, σ)
14: **end for**

図 6　頻出複合構造グラフパターン発見アルゴリズム FMG
Fig. 6　Pseudo code of FMG.

3.4　頻出複合構造グラフパターンの完全列挙

頻出複合構造グラフパターン発見アルゴリズム FMG を**図 6** に示す．FMG-Enum では，標準形を満たすか否かにかかわらず，まず内部拡張を適用する（1–6 行目）．その後は，標準形を満たす頻出パターンを対象に，パターンの出力（8 行目）と外部拡張（9–14 行目）を行う．

アルゴリズム FMG は，(1) 外部拡張が gSpan の最右拡張に，(2) 内部拡張が，前方辺とともに導入される頂点のラベルの考慮にそれぞれ対応しており，gSpan の素直な拡張となっている．したがって，gSpan の完全性より，複合構造グラフデータベースからの頻出パターン発見に対する FMG の完全性が保証される．

4.　内部単調制約と内部飽和性による枝刈りの導入

FMG は，複合構造グラフデータベースからすべての頻出パターンを漏れなく列挙する．しかし，パターンの内部構造として構造データの組合せを考慮するため，列挙されるパターン数は指数的に爆発し，パターンの発見やその後の解析に際して必ずしも現実的ではない．この爆発を防ぐために，本稿では，(1) 全頂点が満たすべき制約を与える，(2) 最右頂点を構成する構造データの各要素を代表的なパターンに限定する，の 2 つの観点からパターンの内部構造に対する制限を加え，列挙するパターンを限定する手法を提案する．

4.1　内部単調制約による枝刈り

複合構造グラフパターン G が制約 C を満たすとき，$C(G)$ と表記する．G が制約 C を満たすとき，その任意の上位グラフ G' もその制約を満たす場合，すなわち $\forall G' \supseteq G\ (C(G) \rightarrow C(G'))$ が成り立つとき，C を**単調制約**と呼ぶ．本稿では，単調制約を**内部単調制約**と**外部単調制約**の 2 つに分ける．外部単調制約は，複合構造グラフパターンの外部構造に制限を与える．たとえば，パターンに含まれなければならない最小の頂点数の要求は，外部単調制約である．これに対し内部単調制約は，頂点の持つ構造データの集合に対して制限を与える．たとえば，頂点の構造データ集合の最小の系列長の要求は，内部単調制約である．

FMG では，外部拡張を用いて，より辺の数の多い特殊なパターンを列挙する．外部拡張を用いることで，外部単調制約を満たさないパターンから制約を満たすパターンが生成されるので，外部単調制約に基づく枝刈りを適用することは困難である．一方，内部単調制約は，効果的な枝刈り手法として適用できる．すなわち，標準形を満たさないパターンと同様，内部単調制約を満たさないパターンに対して，外部拡張の適用は行わない．これは，内部単調制約を満たさないパターンに対する外部拡張が，その後の拡張によって制約を満たすパターンを生成することのできない，最右頂点以外に内部単調制約を満たさない頂点を持つパターンを生成してしまうことによる．したがって，外部拡張はパターンの最右頂点が内部単調制約を満たすパターンに対してのみ適用する．本稿ではこの枝刈りを**内部単調制約による枝刈り**と呼ぶ．

図 7 に，内部単調制約による枝刈りの例を示す．今，「系列の長さが 3 以上である」という内部単調制約を考える．このとき，P_7 は，最右頂点中の系列 $\langle A \rangle$ の長さが 1 なので内部単調制約に違反している．したがって，P_7 に対しては外部拡張はされない．一方，P_7 に対

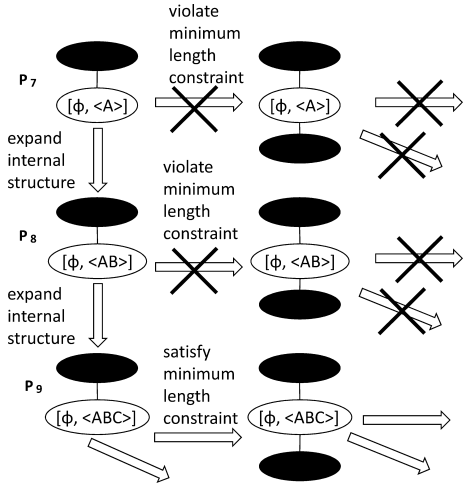


図 7　内部単調制約を用いた拡張
Fig. 7　Expanding with internal monotone constraint.

して内部拡張を適用することで，P_8 を得る．P_8 は，最右頂点中の系列 $\langle AB \rangle$ の長さが 2 なので，内部単調制約を満たさない．したがって，再び内部拡張のみが適用される．一方，得られた P_9 は内部単調制約を満たしており，内部拡張および外部拡張の両方が適用される．

4.2　内部飽和性による枝刈り

ある複合構造グラフパターン p に対し，内部拡張を適用することで得られたパターンを q とする．データベース D 中の各グラフ G に対し，G における p の全出現と G における q の全出現が等しいとき，すなわち，

$$\forall G \in D\ (\Phi_G^p = \Phi_G^q)$$

が成り立つとき，p と q はデータベース D において**内部出現マッチする**といい，$OM_D(p,q)$ と表記する．またこのとき，p と q の支持度は同じとなる．たとえば図 1 において，

$$\begin{aligned}\Phi_{G_1}^{P_1} = \Phi_{G_1}^{P_2} = &\ \{\{(v_{13},v_{14}),(v_{13},v_{11})\}\}\\ \Phi_{G_2}^{P_1} = \Phi_{G_2}^{P_2} = &\ \{\{(v_{21},v_{22}),(v_{21},v_{24})\},\\ &\ \{(v_{22},v_{25}),(v_{22},v_{21})\}\}\end{aligned}$$

であるので，p_1 と p_2 は内部出現マッチする．

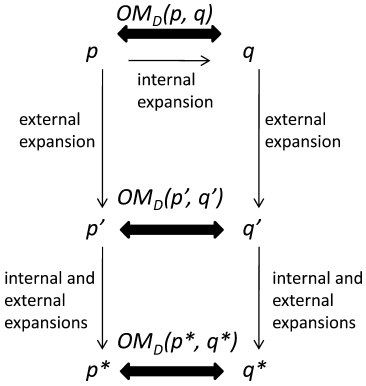


図 8　内部出現マッチと拡張の関係
Fig. 8　The relationship between internal occurrence matching and expansions.

パターン p と q が内部出現マッチするとき，p に辺 e を追加することで得られるパターン p' と，同じく q に e を追加することで得られるパターン q' に対し，$OM_D(p',q')$ が成り立つ（**図 8** 参照）．これは，G をデータベース中のグラフとしたとき，$\Phi_G^p = \Phi_G^q$ より，

$$\forall op \in \Phi_G^p (op \cup \{e\} \in \Phi_G^{p'} \leftrightarrow op \cup \{e\} \in \Phi_G^{q'})$$

が成り立つ，すなわち，p' の出現に含まれる op は q' の出現にも含まれ，p' の出現に含まれない op は q' の出現にも含まれないためである．加えて，p' に対して繰り返し拡張を行うことで得られるパターン p^* に対し，p^* と同じ支持度を持つパターン q^* を，q' に対する拡張の繰返し適用により得ることができる（図 8 参照）．すなわち，パターン x がパターン y に対する内部・外部拡張の繰返し適用により得られることを $x \succ_e y$ と表記すると，

$$\begin{aligned}OM_D(p,q) \rightarrow (\forall p^* \succ_e p'\ (\exists q^* \succ_e q'\\ (sup_D(p^*) = sup_D(q^*) \wedge p^* \subseteq q^*)))\end{aligned}$$

が成り立つ．これも，出現の等価性によるものである．

ところで，p' と q'，p^* と q^* は，それぞれ同じ支持度を持つ．加えて，q' は p' を，q^* は p^* をそれぞれ包含するので，p' や p^* の列挙は冗長であると考えられる．このことに基づき，p と q がデータベースにおいて内部出現マッチするとき，p に対して外部拡張の適用を行わないという枝刈り手法を提案する．本稿ではこの枝刈りを，**内部飽和性による枝刈り**と

Algorithm CCFMG(D, σ, C_{in}, C_{ex})

1: set I be a set of all initial patterns
2: **for each** $p \in I$
3:　**if** $\sup_D(p) \geq \sigma$ **then**
4:　　CCFMG-Enum(p, D, σ, C_{in}, C_{ex})
5: **end for**

Subroutine CCFMG-Enum(p, D, σ, C_{in}, C_{ex})

1: set P_{in} be a set of patterns obtained
2:　by expanding internal structure of p
3: **for each** $s \in P_{in}$
4:　**if** $sup_D(s) \geq \sigma$ **then**
5:　　CCFMG-Enum(s, D, σ, C_{in}, C_{ex})
6: **end for**
7: **if** $\exists s \in P_{in}$ s.t. $OM_D(p, s)$ **then** return
8: **if** $\neg$isCanonical(p) $\vee$ $\neg C_{in}(p)$ **then** return
9: **if** $C_{ex}(p)$ **then output** p
10: scan D and set E be a set of all edges e
11:　s.t. p can be right most extended
12: **for each** $e \in E$
13:　**if** $sup_D(p \cdot e) \geq \sigma$ **then**
14:　　CCFMG-Enum($p \cdot e$, D, σ, C_{in}, C_{ex})
15: **end for**

図 9　単調制約付き頻出複合構造グラフパターン発見アルゴリズム CCFMG
Fig. 9　Pseudo code of CCFMG.

呼ぶ.

4.3　枝刈り手法をともなうパターンの列挙

内部単調制約による枝刈り，および内部飽和性による枝刈りを用いることによって FMG を拡張した CCFMG を**図 9** に示す．図中において，C_{in} と C_{ex} は，内部単調制約と外部単調制約をそれぞれ表す.

CCFMG は，FMG の自然な拡張である．まず，標準形であるか否か，および制約を満たすか否かにかかわらず，内部拡張を適用する（CCFMG-Enum の 1–6 行目）．ついで，内部飽和性による枝刈り（7 行目）および内部単調制約による枝刈り（8 行目）を確認し，いずれの場合でも枝刈りされない標準形パターンに対して，外部拡張を適用する（10–15 行目）.

5. 関 連 研 究

近年，多くのグラフマイニング手法が考案され，現実の問題に適用されている[2),6)–8),11)–13),19),20)]．しかし，頂点内部の構造に着目した手法は少ない.

DAG Miner[3)] は，頂点がアイテム集合からなる有向非巡回グラフ（DAG）データベースから，頻出パターンを発見する手法であり，FMG との関連性が強い．DAG Miner は，頂点の候補となるアイテム集合をあらかじめすべて求めておき，そのアイテム集合を頂点とする pyramid pattern と呼ばれる特別な形状をした頻出パターンを発見する．これに対し，FMG は，内部構造と外部構造を同時に列挙する.

一方，これまでに複雑な木構造パターンを列挙する手法が提案されている．FAT-Miner[10)] は，頂点が属性集合からなる木構造データベースから，頻出パターンを発見する手法である．また，pFreqT[15)] は，頂点が系列からなる木構造データベースから，頻出パターンを発見する手法である．これらの 2 つの手法は，頂点に 1 つの構造データを持つ木構造を扱うのに対して，FMG は，頂点に複数の構造データを持つグラフを対象としている．したがって，FMG はこれらの手法よりも柔軟にパターンを列挙することができるといえる.

また，利用者が積極的に制約を与えるアプローチは，構造データベースからの頻出パターン発見に際しよく使用されている．しかし，グラフ特有の性質に関する制約を与える手法は少ない．gPrune[21)] は，グラフ構造に対して逆単調制約を与えることによる枝刈りを行うことで，制約を満たすグラフパターンの列挙を効率的に行っている．一方，CCFMG では，内部構造に単調制約を与えることにより，パターン探索空間の枝刈りを実現している.

6. 評 価 実 験

提案手法の有効性を評価するため，Java 言語を用いて FMG および CCFMG を実装し，評価実験を行った．評価には，(1) 突然変異誘発性物質データ MUT[9),16)]（データ数 230，平均頂点数 25.62，平均辺数 27.43），および (2) KEGG の代謝パスウェイデータ（データ数 510，平均頂点数 19.20，平均辺数 18.09）の 2 つの実データを用いた.

実験では，最小支持度を徐々に下げながら，実行時間，発見されるパターン数，内部単調制約による枝刈りが適用されたパターン数，内部飽和性による枝刈りが適用されたパターン

表 1　突然変異誘発性物質データを用いた実験結果
Table 1　Experimental results of MUT.

σ[%]	pattern	time [sec]	closed
100	235	2	430
95	1,464	33	2,449
90	6,475	100	10,625
85	37,765	794	64,489
80	131,564	2,271	252,597
75	258,406	3,954	643,986
70	352,133	4,984	1,187,682
65	902,405	10,026	2,445,408
60	1,325,812	14,257	4,401,909
55	2,431,279	43,861	8,582,730

数を計測した．また，すべての実験は，Linux マシン（AMD Opteron 252 2.6 GHz，メインメモリ 2 GB）上で行った．

MUT は，原子，疎水性，電荷の組をアイテム集合とし，アイテム集合を 1 つ持つ構造データ集合を頂点とする複合構造グラフとして表現した．なお，疎水性，電荷は連続値のため，それぞれ 10 種の離散値に変換した．MUT を用いた実験結果を**表 1** に示す．表中で，closed は内部飽和性による枝刈りが適用された回数を表す．なお，本実験においては，内部単調制約は与えていない．FMG は，支持度 100%のときに 1,200 のパターンを約 11 秒で，95%のときに 43,024 のパターンを約 972 秒で発見した．しかし，支持度 90%以下の場合，24 時間以内に結果が得られなかった．これは内部飽和性による枝刈りが適用されず，内部出現マッチするパターンをさらに拡張してしまうので，同一の探索空間上を何度も探索しているためであると考えられる．これに対し CCFMG は，内部飽和性による枝刈りにより，より低支持度のパターンの抽出に成功している．これらのことから，内部飽和性による枝刈りの効果が確認できる．

代謝パスウェイデータは，KEGG 上で公開されている 510 種の各生物について，リボフラビン代謝と呼ばれる生物のビタミン B_2 に関する代謝パスウェイを複合構造グラフで表現したものである．このパスウェイの頂点は，化合物，酵素，パスウェイのリンクのいずれかから構成されている．化合物は，化合物名を持ち，アイテムとして扱う．また，この化合物が出現しているパスウェイの集合の情報を持ち，これをアイテム集合とする．酵素は，酵素の種別を 4 段階の階層に分類する ECNumber と呼ばれる数値が割り当てられており，階層構造として扱う．また，酵素にはアミノ酸の配列が含まれており，これを系列データとする．

表 2　代謝パスウェイを用いた実験結果
Table 2　Experimental results of metabolic pathways.

σ[%]	pattern		time [sec]		closed	monotone
	FMG	CCFMG	FMG	CCFMG	CCFMG	CCFMG
100	1	1	0	0	0	0
95	63	1	19	6	0	21
90	380	4	63	11	0	134
85	-	6	-	15	0	258
80	-	11	-	15	0	302
75	-	32	-	19	5	409
70	-	37	-	20	5	503
65	-	63	-	24	6	650
60	-	164	-	31	22	869
55	-	456	-	49	86	1,337
50	-	921	-	65	171	1,779
45	-	2,123	-	89	396	2,557
40	-	4,301	-	149	850	3,126
35	-	4,567	-	288	956	5,383
30	-	5,098	-	558	1,277	10,952
25	-	7,835	-	1,646	3,132	30,151
20	-	31,941	-	7,426	22,045	138,628
15	-	364,460	-	101,730	347,666	1,729,506

系列の長さは生物種，酵素によって異なるが，1,000 以上になる系列も多い．パスウェイのリンクはアイテムとして扱う．よって，パスウェイの頂点内部の構造データ集合を，[アイテム，階層構造，アイテム集合，系列] と定義した．また，CCFMG を適用する際に，系列長が 8 以上でなければならないという内部単調制約を与えた．本来は全頂点が内部単調制約を満たさなければならないが，系列データを持つのは酵素のみであるので，系列を持たない頂点を含むパターンの列挙を許容している．

代謝パスウェイを用いた実験結果を**表 2** に示す．表中において，monotone は内部単調制約による枝刈りが適用された回数である．MUT での実験同様，FMG では，支持度 85%以下の場合に 48 時間以内に結果が得られなかった．一方，CCFMG は，内部単調制約，および内部飽和性制約により，より低い支持度を持つ頻出パターンの発見に成功している．また，発見されるパターンの系列長を 8 以上に限定していることで，系列長の短いパターンの発見を抑圧している．**図 10** に発見された代謝パスウェイのパターンの例を示す．このパターンは，頂点が各構造データから構成されており，従来手法では発見することができない．また，閉路を含むグラフ構造となっている．このことからも，FMG や CCFMG が有

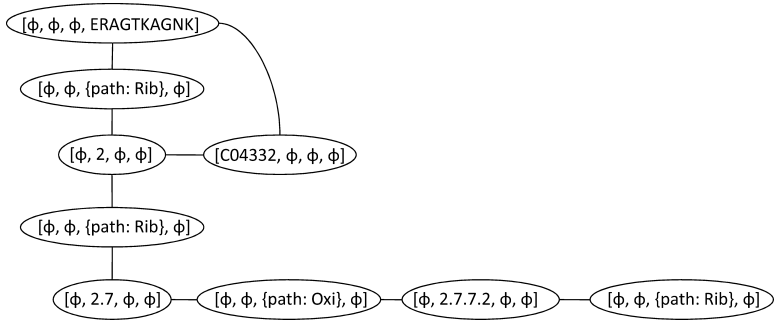


図 10　発見されたパターンの例
Fig. 10　An example of discovered pattern.

効であることが確認できる．

7. ま と め

本稿では，頻出複合構造グラフパターンの発見問題に対し，内部構造と外部構造の拡張によるパターンの列挙アルゴリズム FMG を提案した．さらに，候補パターンの削減を目的とする内部飽和性と内部単調制約を用いた新しい枝刈り技法を提案した．

今後の課題としては，(1) 支持度とは別の基準を用いることにより，より意味のあるパターンを列挙する，(2) 複合構造グラフの外部構造に制約を与えることなどにより利用者が理解しやすい結果を得る，(3) 出現を効率良く更新する，データベースを制限する手法などを適用することによって，速度の向上を試みることがあげられる．

謝辞　本研究の一部は，文部科学省科学研究費補助金（若手研究（B）：課題番号 19700146，特定領域研究：課題番号 19024055，基盤研究（B）：課題番号 20300038）による．

参 考 文 献

1) Borgelt, C.: On Canonical Forms for Frequent Graph Mining, *Proc. 3rd International Workshop on Mining Graphs*, pp.1–12 (2005).
2) Borgelt, C. and Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules, *Proc. 2nd IEEE International Conference on Data Mining*, pp.51–58 (2002).
3) Chen, Y.L., Kao. H. and Ko, M.: Mining DAG Patterns from DAG Databases, *Proc. 5th International Conference on Web-Age Information Management*, pp.579–588 (2004).
4) De Raedt, L., Washio, T. and Kok, J.N. (Eds.): *Advances in Mining Graphs, Trees and Sequences*, Volume 124 Frontiers in Artificial Intelligence and Applications, IOS Press (2005).
5) Han, J. and Kamber, M.: *Data Mining: Concepts and Techniques*, Morgan Kaufmann (2005).
6) Huan, J., Wang, W. and Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism, *Proc. 3rd IEEE International Conference on Data Mining*, pp.549–552 (2003).
7) Inokuchi, A., Washio, T. and Motoda, H.: An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data, *Proc. 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp.13–23 (2000).
8) Inokuchi, A., Washio, T. and Motoda, H.: Complete mining of frequent patterns from graphs: Mining graph data, *Machine Learning*, Vol.50, pp.321–354 (2003).
9) King, R.D., Srinivasan, A. and Sternberg, M.J.E.: Relating chemical activity to structure: An examination of ILP successes, *New Generation Computing*, Vol.13, No.3-4, pp.411–433 (1995).
10) Knijf, J.D.: FAT-miner: Mining Frequent Attribute Trees, *Proc. 2007 ACM symposium on Applied computing*, pp.417–422 (2007).
11) Koyuturk, M., Grama, A. and Szpankowski, W.: An efficient algorithm for detecting frequent subgraphs in biological networks, *Bioinformatics*, Vol.20, pp.200–207 (2004).
12) Kuramochi, M. and Karypis, G.: Frequent subgraph discovery, *Proc. 1st IEEE International Conference on Data Mining*, pp.313–320 (2001).
13) Nijssen, S. and Kok, J.N.: A Quickstart in Frequent Structure Mining can make a Difference, *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.647–652 (2004).
14) Pei, J., Han, J., Mortazavi-Asl, B., Pnto, H., Chen, Q., Dayal, U. and Hsu, M.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, *Proc. International Conference on Data Engineering*, pp.215–224 (2001).
15) Sato, I. and Nakagawa, H.: Semi-structure Mining Method for Text Mining with a Chunk-Based Dependency Structure, *Proc. 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.777–784 (2007).
16) Srinivasan, A., Muggleton, S., King, R. and Sternberg, M.J.E.: Mutagenesis: ILP experiments in a non-determinate biological domain, *Proc. 4th International Workshop on Inductive Logic Programming* (*ILP'94*), pp.217–232 (1994).
17) Uno, T., Asai, T., Uchida, Y. and Arimura, H.: An Efficient Algorithm for Enumerating Closed Patterns in Transaction Databases, *Proc. 7th International Conference*

on Discovery Science, pp.16–31 (2004).
18) Washio, T. and Motoda, H.: State of the art of graph-based data mining, *SIGKDD Explorations*, Vol.5, No.1, pp.59–68 (2003).
19) Yan, X. and Han, J.: CloseGraph: Mining Closed Frequent Graph Patterns, *Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.286–295 (2003).
20) Yan, X. and Han, J.: gSpan: Graph-Based Substructure Pattern Mining, *Proc. 2nd IEEE International Conference on Data Mining*, pp.721–724 (2002).
21) Zhu, F., Yan, X., Han, J. and Yu, P.S.: gPrune: A Constraint Pushing Framework for Graph Pattern Mining, *Proc. 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp.388-400 (2007).

(平成 19 年 12 月 20 日受付)
(平成 20 年 2 月 13 日採録)

(担当編集委員　岩井原 瑞穂)

山本　翼

1984 年生．2007 年神戸大学工学部情報知能工学科卒業．現在，神戸大学大学院工学研究科に在学中．構造データマイニングの研究に従事．

尾崎　知伸

1973 年生．1996 年慶應義塾大学総合政策学部卒業．1998 年同大学大学院政策・メディア研究科前期修士課程修了．2002 年同研究科講師．2005 年神戸大学大学院自然科学研究科助手．現在，神戸大学自然科学系先端融合研究環助教．博士（政策・メディア）．帰納論理プログラミング，構造データマイニング等の研究に従事．人工知能学会会員．

大川　剛直（正会員）

1963 年生．1986 年大阪大学工学部通信工学科卒業．1988 年同大学大学院工学研究科通信工学専攻博士前期課程修了．大阪大学助手，講師，助教授を経て，2005 年神戸大学大学院自然科学研究科教授．現在，神戸大学大学院工学研究科教授．博士（工学）．知的ソフトウエア，バイオインフォマティクス等の研究に従事．IEEE，人工知能学会，電子情報通信学会，電気学会等の会員．