社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－DBS－143 (79)
2007／7／4

ラベル付きグラフのフィルタリングのための行列サイズ縮小手法

長屋 未来† 片山 薫†

† 首都大学東京システムデザイン研究科 〒191-0065 東京都日野市旭が丘6-6
E-mail: †mirai.nag@gmail.com, ††kaoru@comp.metro-u.ac.jp

あらまし 画像における特徴量抽出，化学式における類似構造の検出などに部分グラフ同型判定は応用できる．しかし部分グラフ同型判定問題はNP完全であり，大規模なグラフを扱う場合，実用時間内に解くことが困難である．我々は部分グラフ同型判定の前処理として，対称行列の固有値に関する事実 (Cauchy の interlace 定理) を利用することを提案し，その有効性を確認した．固有値計算に必要な行列サイズを縮小する事で interlace 定理を利用するためのコストを減らし，interlace 定理の判定精度を改善する事は有効であると考えられる．本稿では，ラベルを利用してグラフをフィルタリングすると共に，グラフを表現する行列のサイズを縮小する手法を提案する．

キーワード 部分グラフ同型，Interlace 定理，固有値，ラベル付きグラフ

Reduction of Matrix Order for Filtering Labeled Graphs

Hideki NAGAYA† and Kaoru KATAYAMA†

† Graduate School of System Design, Tokyo Metropolitan University Asahigaoka 6-6, Hino-shi, Tokyo, 191-0065 Japan
E-mail: †mirai.nag@gmail.com, ††kaoru@comp.metro-u.ac.jp

Abstract It is often used to solve the subgraph isomorphism problem in various application such as retrieval for a chemical formula, feature extraction from pictures. This problem is NP-complete and require large computation cost to solve it. We suggested to use the fact concerning the symmetric eigenvalue (Cauchy's interlace theorem) as a preprocessing of the subgraph isomorphism. It is effective to reduce order of the matrix for decreasing the cost to calculate eigenvalue. In this paper, we suggest a method reducing order of the matrix while filtering labeled graphs by using label of graphs.

Key words Subgraph Isomorphism,Interlace Theorem,Eigenvalue,Labeled Graph

1. はじめに

グラフは画像検索 [10] や化合物の検索 [8]，パターン認識など幅広い分野で応用されている．2つのグラフが与えられたとき，一方のグラフ G_l が他方のグラフ G_s を含んでいるかを調べる問題（部分グラフ同型判定問題）がある．この問題はNP完全の問題であるため，大きなグラフを扱う場合，計算コストが膨大になる．我々は部分グラフ同型判定の前処理として，対称行列の固有値に関する事実 (interlace 定理) を利用して部分グラフではないものをフィルタリングする手法 [9] を提案し，以下の場合に有効である事を確認した．(G_l における頂点，枝の集合を V_l, E_l とし，G_s における頂点，枝の集合を V_s, E_s とする．)

- G_s が G_l の誘導部分グラフかどうかを判定する場合，グラフ G_l の枝が密であり，グラフ G_s の枝が疎である．$|V_s| \leq |V_l|$ であり，その頂点数の差が小さいこと．
- G_s が G_l の誘導部分グラフかどうかを判定する場合，グラフにラベルを使用せず，$|V_s| \leq |V_l|$ であり，$|E_s| \leq |E_l|$ である．その頂点数，枝数共に差が小さいこと．

固有値計算に必要な行列サイズを縮小する事で interlace 定理を利用するためのコストを減らし，interlace 定理の判定精度を改善する事ができる．本稿では，G_s に使用されているラベルが G_l に全て含まれている事を利用して，部分グラフではないグラフをフィルタリングすると共に，interlace 定理を利用する上で必要ない枝，頂点をラベルにより削る事で行列サイズを縮小する手法を提案する．実験により，G_s に使用されているラベルが G_l に使用されているラベルより少なく，$|V_l|$, $|V_s|$ の差が少なく，$|E_s|$ より $|E_l|$ が大きい場合，提案手法が有効である事が確認できた．

2. 関連研究

部分グラフ同型判定問題を組み合わせ的な手法で解くアルゴリズムとしては，VF2 [2] や Ullmann による手法 [1] がある．バックトラック法を使って検索範囲を減少させて，グラフ同型判定と部分グラフ同型判定を効率よく行う手法が Ullmann の

手法である．VF2は，深さ優先探索に基づく手法で，大規模で複雑なグラフに対してグラフ同型判定，部分グラフ同型判定を行うことができる．Yanら[4]は，グラフデータベース内の頻出するグラフや特徴のあるグラフを検索し，索引として使用するgIndexと呼ばれる手法を提案した．Yanら[7]は，頻出するグラフや特徴のあるグラフを使ってフィルタリングを行い，グラフの類似検索を行う手法を提案した．グラフ構造データ中に現れる特徴的なパターンを抽出するために，二つのノードおよびそれらをつなぐリンクの逐次抽出を行うYoshidaら[5]の手法がある．

3. 定　　義

3.1 コーシーのinterlace定理

式(1)で表される次数がnの対称行列Aを考える．Hは，Aの主部分行列であり，その次数は$m(n > m)$であるとする．

$$A = \begin{pmatrix} H & B \\ B^* & C \end{pmatrix} \tag{1}$$

対称行列Aの固有値を$\alpha_i(i = 1,2,\ldots,n,\ \alpha_1 \leqq \alpha_2 \leqq \ldots \leqq \alpha_n)$，主部分行列$H$の固有値を$\theta_j(j = 1,2,\ldots,m,\ \theta_1 \leqq \theta_2 \leqq \ldots \leqq \theta_m)$とするとき，$H$が対称行列$A$の主部分行列であれば，全ての$k(k = 1,2,\cdots m)$について次式で表される不等式が成り立つ．

$$\alpha_k \leqq \theta_k \leqq \alpha_{k+(n-m)} \qquad (k = 1,2,\cdots m) \tag{2}$$

以降，全てのkについて式(2)の不等式が成り立つ事を「interlace定理を満たす」と呼び，1つでも不等式が成り立たない場合を「interlace定理を満たさない」と呼ぶ．

3.2 グラフの定義

定義1 提案手法はラベル付きグラフを対象とし，L_V, L_Eをそれぞれ頂点，枝のラベルの集合とする．ラベルつきグラフは4つの要素$G = (V, E, \mu, \nu)$から成る．頂点をvとし，頂点の集合をV，枝をeとし，枝の集合を$E \subseteqq V \times V$とする．ve, weはeの両端にある頂点とする．$\mu : V \rightarrow L_V, \nu : E \rightarrow L_E$はそれぞれ頂点，枝にラベルを与える関数である．以降，本稿ではラベル付きグラフをグラフと呼ぶ．

定義2 $G_s = (V_s, E_s, \mu_s, \nu_s)$が$G_l$の部分グラフであるとき，そのときに限り以下の条件を満たす．また，G_sがG_lの部分グラフ($G_s \subseteqq G_l$と表記し，G_lはG_sの親グラフともいう)であるとは，以下を満たすことをいう．

- $V_s \subseteqq V_l$
- $\forall v_s \in V_s, \mu_l(v_s) = \mu_s(v_s)$
- $E_s \subseteqq E_l \cap vewe \in E_s, ve, we \subseteqq V_s$
- $\forall e_s \in E_s, \nu_l(e_s) = \nu_s(e_s)$

$G_s \subseteqq G_l$であり，かつveとweを接続する枝が$vewe \in E_s$であるとき，G_sをG_lの誘導部分グラフという．

3.3 グラフの接続行列表現

3.3.1 隣接行列

$V = \{v1,\ldots,vn\}$と$vivj \in E$をもつグラフ$G = (V, E, \mu, \nu)$に対して，隣接行列$A = (a_{ij})_{n \times n}$を以下のように定義する．

$$A_{ii} = \mu(vi) \tag{3}$$

$$A_{ij} = \begin{cases} \nu(vivj) & (vivj \in E \text{のとき}) \\ 0 & (\text{そうでないとき}) \end{cases} \tag{4}$$

頂点のラベルの値を対角要素へ入れ，枝のラベルの値をを非対角要素に入れる．隣接行列は，対称行列であり，その次数は頂点数と同じである．

3.3.2 接続行列

$V = \{v1,\ldots,vm\}$と$vivj \in E$をもつグラフ$G = (V, E, \mu, \nu)$に対して，接続行列$N = (b_{ij})_{m \times n}$を以下のように定義する．

$$N_{ij} = \begin{cases} 1 & (vi \text{が} vivj \in E \text{に接続されているとき}) \\ 0 & (\text{そうでないとき}) \end{cases} \tag{5}$$

interlace定理を利用する場合，対称行列にしなければならない．そこで対称行列を得るために，Haemers[6]の接続行列Nを用いた対称行列表現(6)を利用する．

$$M = \begin{pmatrix} 0 & N \\ N^t & 0 \end{pmatrix} \tag{6}$$

我々は，式(6)に以下を定義[9]することで頂点，枝のラベルを利用する．

$$M_{ii} = \begin{cases} \mu(v_i) & (1 \leqq i \leqq m) \\ \nu(e_{i-m}) & (m+1 \leqq i \leqq m+n) \end{cases} \tag{7}$$

この行列の次数は，グラフの頂点数と枝数の和($|V|+|E|$)になる．以降本稿では，式(6)，式(7)の形を接続行列と呼ぶ．

4. 提案手法

提案手法は，グラフG_sがグラフG_lの部分グラフであるかどうかを判定するために利用される．組み合わせ的なアルゴリズムで部分グラフ同型判定を行う前に，ラベルによるフィルタリングとInterlace定理によるフィルタリングを行う．

4.1 定　　義

グラフに使用されているラベルは頂点，枝を問わず数値とする．グラフG_xの頂点をv_x，頂点の集合をV_x，枝を$e_x = (vex, wex)$，枝の集合をE_xで表す．(vex, wex)は，e_xの両端に接続されている頂点とする．本稿では頂点，枝のラベルに数値を使う．頂点,枝には，IDが割り当てられているものとする．提案手法で使用する記号を以下に定義する．以下は，グラフG_xについて定義し，G_l, G_sについても同様に定義する．

- IDの値がyである枝をe_{xy}と表現する．
- vex, wexについて，$\mu_x(vex)$,$\mu_x(wex)$の2つの値を比較し，小さい方の値をl_{vex}，大きい方の値をl_{wex}とする．
- e_xを入力したときにl_{vex}, l_{wex},$\nu_x(e_x)$を与える関数をξとする．$\xi : e_x \rightarrow l_{vex} \times l_{wex} \times \nu_x(e_x)$

- G_x に存在する枝の接続がない頂点 (non-edge な頂点) を vne_x と書く．また vne_x の集合を Vne_x とする．
- Vne_x において，頂点のラベル z と同じ値をもつ頂点の総数を o_{xz} で表し，その集合を O_x とする．
- $\mu_x(vne_x)$ と Vne_x を入力すると o_{xz} を与える関数を ι とする．$\iota : \mu_x(vne_x) \times Vne_x \rightarrow o_{xz}$

グラフ G_l, G_s について，以下を定義する．

- E_l, E_s を入力すると，$\xi(e_s) = \xi(e_l)$ となる全ての $e_l \in E_l$ を集合 R として与える関数を κ とする．$\kappa : E_l \times E_s \rightarrow R$
- $o_{sz} - o_{lz}$ の値を d_z とする．値が負となった場合，d_z は 0 とする．d_z の集合を D とする．
- Vne_x, R, D を利用して，G_l, G_s から interlace 定理を利用する上で必要のない頂点，枝を除いたグラフを G'_l, G'_s とする．
- グラフ G_l, G_s, G'_l, G'_s の行列表現を g_l, g_s, g'_l, g'_s とする．
- 行列 g'_l の固有値を $\alpha_i (i = 1, 2, \ldots n, \alpha_1 < \alpha_2 < \ldots < \alpha_n)$ とし，行列 g'_s の固有値を $\theta_j (j = 1, 2, \ldots m, \theta_1 < \theta_2 < \ldots < \theta_m)$ とする．n は行列 g'_l の次数，m は行列 g'_s の次数とする．

4.2 グラフの入力形式

グラフ G_x について，以下を入力する．G_l, G_s を G_x と同様に入力する．

- 頂点 v_x の ID，$\mu_x(v_x)$
- 枝 e_x の ID，$\nu_x(e_x)$，e_x に接続されている頂点 (vex, wex) の ID

図 1 の G_s の入力例を挙げる．入力された各値は，以下の配列に保存する．

- $V_x = \{1, 2, 3, 4\}$(頂点の ID)
- $L_{Vx} = \{1, 2, 4, 1\}$(頂点のラベル)
- $E_x = \{1, 2\}$(枝の ID)
- $L_{Ex} = \{1, 3\}$(枝のラベル)
- $Vex = \{1, 2\}$(接続されている頂点の ID)
- $Wex = \{2, 3\}$(接続されている頂点の ID)

データはすべて配列で管理する．配列 V_x, L_{Vx} の次数は同じであり，配列 E_x, L_{Ex}, Vex, Wex の次数は同じである．以下において，a は，配列におけるポインタを表す．頂点について，V_x の a 番目の頂点は，L_{Vx} の a 番目のラベルを持つ．枝について，E_x の a 番目の枝は，L_{Ex}, Vex, Wex の a 番目のラベル，接続されている頂点を持つ．入力例では，ID が 3 である頂点のラベルは 4 であり，ID が 2 である枝のラベルは 3，接続されている頂点の ID が 2 と 3 である．

4.3 ラベルによるフィルタリング

ラベルによるグラフのフィルタリングは，$\nu_x(e_x)$, l_{vex}, l_{wex} で行う．枝のラベル $\nu_x(e_x)$ と枝の両端に接続されている頂点のラベル (l_{vex}, l_{wex}) を使用して，すべての $e_s \in E_s$ について，$\xi(e_s) = \xi(e_l)$ であるかを調べる．もし，すべての $e_s \in E_s$ について，$\xi(e_s) = \xi(e_l)$ でなければ，部分グラフではないと判定する．図 1 を例に考える．図中の id と L はそれぞれ枝の ID とラベルであり，頂点の中の数値は頂点のラベルである．図 1 の G_l と G_s は，部分グラフ同型である．

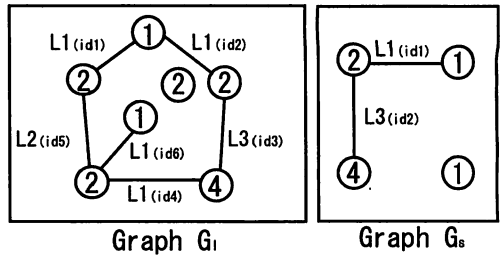

図 1 グラフの例

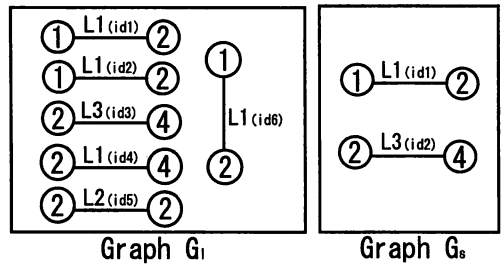

図 2 $\xi(e)$ によるイメージ

全ての $e_s \in E_s$ について，$\xi(e_s) = \xi(e_l)$ を調べる．すべての $e_s \in E_s$ と $e_l \in E_l$ について，$\xi(e_l)$,$\xi(e_s)$ を図にしたものを図 2 に示す．G_s には枝が 2 つ存在し，$\xi(e_s) = (l_{ves}, l_{wes}, \nu_s(e_s))$ は，$\xi(e_{s1}) = (1, 2, 1)$,$\xi(e_{s2}) = (2, 4, 3)$ の 2 つである．E_l には，(1, 2, 1), (2, 4, 3) が含まれている ($\xi(e_{l1})$,$\xi(e_{l3})$ 等) ので，すべての $e_s \in E_s$ について，$\xi(e_s) = \xi(e_l)$ である．G_l が G_s の親グラフであれば，すべての $e_s \in E_s$ について $\xi(e_s) = \xi(e_l)$ であるが，すべての $e_s \in E_s$ について $\xi(e_s) = \xi(e_l)$ であっても，G_l が G_s の親グラフとは限らない．

4.4 行列サイズの縮小

グラフを接続行列によって表現すると，次数が $|V| + |E|$ となる．以下によって，interlace 定理を利用する上で必要の無い枝や頂点を G_l, G_s から除き，行列の次数を小さくする．行列の次数を小さくすると，固有値計算のコストを減らすことができる．以下の (1) は，図 9 の処理 5 に対応し，(2) は処理 6，(3) (4) は処理 7 に対応する．

(1) $\kappa(E_l, E_s) \rightarrow R$

(2) 集合 R から，枝の接続がない頂点 Vne_l を得る．また，E_s から，枝の接続がない頂点 Vne_s を得る

(3) $\iota : \mu_s(vne_s) \times Vne_s \rightarrow o_{sz}$，$\iota : \mu_l(vne_l) \times Vne_l \rightarrow o_{lz}$，$o_{lz} \leqq o_{sz}$ ならば，$d_z = 0$．そうでなければ，$d_z = o_{sz} - o_{lz}$．

(4) 行列 g'_s を作るとき，ラベル z を持つ枝の接続がない頂点 Vne_s は，d_z 個使用する．枝の接続がない頂点 Vne_l は，行列 g'_l に使用しない

図 1 を使って例を示す．図 2 から，$\xi(e_s)$ は，$\xi(e_{s1}) = (1, 2, 1)$,$\xi(e_{s2}) = (2, 4, 3)$ の 2 つがある．$\xi(e_{s1}) = (1, 2, 1)$ と同じ値を持つ枝を G_l から求めると，$\xi(e_{l1})$,$\xi(e_{l2})$,$\xi(e_{l6})$ の 3 つが求められる．同様に，$\xi(e_{s2}) = (2, 4, 3)$ と同じ値をもつ枝は

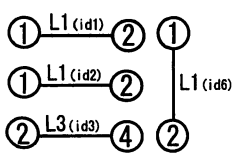

図 3 $\xi(e_s) = \xi(e_l)$

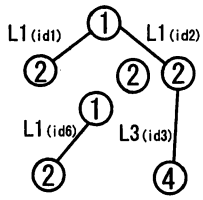

図 4 グラフ G_l''

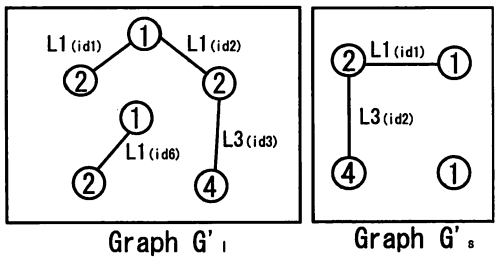

図 5 グラフ G_l', G_s'

$\xi(e_{l3})$ である．$\xi(e_s) = \xi(e_l)$ によって求められた e_l を図 3 に示す．G_l が G_s の親グラフであるためには，図 3 の枝 e_{l1}, e_{l2}, e_{l3}, e_{l6} の枝が必要である．G_l から e_{l4}, e_{l5} の枝を除く．e_{l4}, e_{l5} をグラフ G_l から除いたグラフ G_l'' を図 4 に示す．

次に G_l'', G_s における枝の接続がない頂点 Vne_l, Vne_s を調べる．G_l'' において，枝の接続がない頂点のラベル $\mu_l(vne_l)$ は，"2" のみで，その総数 o_{l2} は 1 個である．G_s において，枝の接続がない頂点のラベル $\mu_s(vne_s)$ は "1" のみで，その総数 o_{s1} は 1 個である．全ての z における o_{sz}, o_{lz} について，その大小関係を調べる．$o_{s1}(=1) > o_{l1}(=0)$ なので，$d_1 = 1$ となる．ラベル "1" の頂点は G_s' に 1 個利用する．$o_{s2}(=0) \leqq o_{l2} = 1$ なので，$d_2 = 0$ となる．ラベル "2" を持つ頂点は，G_s', G_l' に使用しない．図 1 から，$\xi(e_s)$,$\xi(e_l)$ と o_{lz}, o_{sz} によって頂点，枝を除いたグラフを図 5 に示す．G_l は G_l' の親グラフであり，G_l' は G_s' の親グラフ ($G_s' \subseteqq G_l' \subseteqq G_l$) になる．(図 1 の例では，頂点を除く事が出来ないので，$G_s = G_s'$ となる) 図 5 の G_l', G_s' を行列で表現することで interlace 定理を利用する．グラフを隣接行列で表現する場合，除いた頂点の数だけ行列を小さく出来る．枝を除くことは，非対角成分の値を $\nu(e) \to 0$ にする．図 6, 7, 8 にグラフ G_l, G_s, G_l' の隣接行列 g_l, g_s', g_l' を示す．g_l' は，g_l の 7 番目の行と列を除き，g_l の (3, 5), (5, 3), (5, 6), (6, 5) 成分を 0 にすることで得られる．g_l' は，g_l よりも次数 1 だけ小さい行列である．g_l' の 5 番目，6 番目の行と列を除くことで g_s' と同じ行列ができる．G_s' が G_l' の誘導部分グラフならば，g_s' は g_l' の主部分行列となり，interlace 定理を満たす．よって，G_l', G_s' を隣接行列で表現することによって，interlace 定理を用いた誘導部分グラフのフィルタリングを行うことができる．同様に，接続行列ついても，$G_s \subseteqq G_l'$ ならば，g_s' は g_l' の主部分行列になる．

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 4 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 2 & 2 & 0 \\ 1 & 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

図 6 隣接行列 g_l

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 3 & 0 \\ 0 & 3 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

図 7 隣接行列 $g_s(=g_s')$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 3 & 4 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

図 8 隣接行列 g_l'

提案手法の処理手順

処理 1: G_l, G_s を入力する
処理 2: 頂点，枝の ID でソートする
処理 3: L_{vex}, L_{wex} を得る
　ves, wes と同じ ID の頂点 v_s にフラグを立てる
処理 4: vex, wex,$\nu_x(e_x)$, l_{vex}, l_{wex} について
　$\nu_x(e_x)$, l_{wex}, l_{vex} の順に安定ソートする
処理 5: $\xi(e_s) = \xi(e_l)$ であるか調べる
　$\xi(e_s) = \xi(e_l)$ となる枝 e_l の ID を R に保存する
処理 6: 枝の接続がない頂点の集合 Vne_s を得る
　R に保存された ID から，e_l を得て，vel, wel によって枝の接続がない頂点 Vne_l を得る
処理 7: Vne_x, R, D によって，G_l, G_s から頂点と枝を除く G_l', G_s' を作成し，G_l', G_s' を行列で表現する
処理 8: 行列 g_l', g_s' の固有値を求める
処理 9: Interlace 定理によるフィルタリングを行う
処理 10: 欲張り法に基づく部分グラフ同型判定を行う

図 9 処理の流れ

処理 3　l_{vex}, l_{wex} =loadLabel(vex, wex)

1: $l_{vex} = \mu_x(wex)$, $l_{vex} = \mu_x(wex)$
2: $l_{vex} \leqq l_{wex}$ となるように l_{vex}, l_{wex} を入れ替える
3: $\mu_s(vex)$, $\mu_s(wes)$ によって 1 度でも使用された頂点 v_s を記録する

図 10 頂点ラベル L_{vex}, L_{wex} を得る

4.5 処理の流れ

部分グラフ同型判定を行う処理は，図 9 で行う．この節において，G_x の処理は，G_l, G_s の両方について行うものとする．

各処理について説明する．入力形式に従って，グラフの頂点 v_x の ID，頂点のラベル $\mu_x(v_x)$，枝 e_x の ID，枝のラベル $\mu_x(v_x)$，接続されている頂点 vex, wex の ID を入力する．処理 3 は，枝 e_x に接続されている頂点 vex, wex のラベルを得るための処理である．l_{vex}, l_{wex} が枝の両端に接続されている頂点のラベルであるが，$l_{vex} \leqq l_{wex}$ となるように入れ替えを行

処理 5　R =Filter(E_s, E_l)

1: $rid \leftarrow 1$
2: すべての $e_s \in E_s$ について，$\xi(e_s) = \xi(e_l)$ ならば
　$\xi(e_s) = \xi(e_l)$ となる $e_l \in E_l$ をすべてもとめ，
　e_l の ID を $r[rid]$ に保存した後，$rid \leftarrow rid + 1$
3: そうでなければ
　部分グラフでは無いと判定して処理を終了する

図 11　枝のラベルによるフィルタリング

処理 7-1　D =non-edgeVertex(Vne_l, Vne_s)

1: 配列 d を 0 で初期化する
2: $\iota(\mu_l(Vne_l) \times Vne_l) \rightarrow O_l$,$\iota(\mu_s(Vne_s) \times Vne_s) \rightarrow O_s$
3: $o_{sz} \leqq o_{lz}$ ならば，$d_z = 0$
4: そうでなければ，$d_z = o_{sz} - o_{lz}$

図 12　D を 得 る

処理 7-2　NID =LargeMatrixID(R)

1: $ctr \leftarrow 0$
2: 次数 $|V_l|$ の配列 NID を-1 で初期化する
3: R に保存された ID によって，枝 e_l の ID を得て，vex, wex を添え字として NID にアクセスする
4: $NID = -1$ ならば
　$NID \leftarrow ctr$ の後，$ctr \leftarrow ctr + 1$

図 13　行列サイズ縮小のための ID 変換配列

処理 7-3　$NIDs$ =smallMatrixID(Ves, Wes)

1: $ctrs \leftarrow 0$
2: 次数 $|V_s|$ の配列 $NIDs$ を-1 で初期化する
3: ves, wes を添え字として $NIDs$ にアクセスする
4: $NIDs = -1$ ならば
　$NIDs \leftarrow ctrs$ の後，$ctrs \leftarrow ctrs + 1$

図 14　G_s のための ID 変換配列

処理 7-4　A =incidenceG_l(NID, R, V_l, E_l, Vel,Wel)

処理 7-2 の後に行い，行列の要素を $A_{i,j}$ で表す
($i, j = 1, 2, \ldots, ctr + |R|$)

1: 次数 $ctr + |R|$ の正方行列 A を 0 で初期化する
2: $NID[v_l] \neq -1$ ならば
　$A_{NID[v_l],\ NID[v_l]} \leftarrow \mu_l(v_l)$
3: R 保存された ID で e_l を得て，vel, wes を添え字として，行列の各成分を求める
　(a) $A_{(ctr+rid),\ (ctr+rid)} \leftarrow \nu_l(e_l)$
　(b) $A_{NID[vel],(ctr+rid)} \leftarrow 1$，$A_{NID[wel],\ (ctr+rid)} \leftarrow 1$
　　$A_{(ctr+rid),\ NID[vel]} \leftarrow 1$，$A_{(ctr+rid),\ NID[wel]} \leftarrow 1$

図 15　接続行列 g'_l

う．G_s の処理について，$\mu_s(ves)$,$\mu_s(wes)$ の処理によって，1 度でも使用された頂点 v_s を記録する．$\mu_s(ves)$,$\mu_s(wes)$ の処理で 1 度も使用されなかった頂点があれば，その頂点はグラフに存在する枝の接続がない頂点である事が分かる．これらの手順

処理 7-5　B =incidenceG'_s($NIDs$, V_s, E_s, Ves, Wes)

処理 7-1，7-3 の後に行い，行列の要素を $B_{i,j}$ で表す
($i, j = 1, 2, \ldots, ctrs + |e|$)

1: 次数 $ctrs + |E_s|$ の正方行列 B を 0 で初期化する
2: $NID[v_s] \neq -1$ ならば，$B_{NID[v_s],\ NID[v_s]} \leftarrow \mu_s(v_s)$
3: e_s の ID に対応した ves, wes によって各成分を求める
　(a) $B_{(ctrs+e_s),\ (ctrs+e_s)} \leftarrow \nu_s(e_s)$
　(b) $B_{NID[ves],(ctr+e_s)} \leftarrow 1$，$B_{NID[wes],\ (ctr+e_s)} \leftarrow 1$
　　$B_{(ctr+e_s),\ NID[ves]} \leftarrow 1$，$B_{(ctr+e_s),\ NID[wes]} \leftarrow 1$
4: $k \leftarrow ctrs + |E_s| + 1$
　for $p \leftarrow 1$ **to** $|D|$ **do**
　　for $q \leftarrow 1$ **to** d_p **do**
　　$A_{k,k} \leftarrow p$ の後，$k \leftarrow k + 1$

図 16　接続行列 g'_s

処理 9　$flag$ =interlaceTheorem(α, n, θ, m)

1: **for** $i = 1$ *to* m **do**
2:　　もし，$\alpha_i \leqq \theta_i \leqq \alpha_{i+(n-m)}$ でなければ
　　部分グラフではないと判定して処理を終了する

図 17　interlace 定理によるフィルタリング

を図 10 に示す．処理 4 によるソートは，処理 5 を行う際，重複してラベルを調べないようにするための処理である．処理 5 は $\xi(e_x)$ よってフィルタリングを行う処理である．l_{vex}, l_{wex}, $\nu_x(e_x)$ の 3 つの値が一致している事を調べる．ラベルを調べると共に，行列サイズを小さくするために必要な $\xi(e_s) = \xi(e_l)$ を調べる．$\xi(e_s) = \xi(e_l)$ となった場合の e_l は，行列 g'_l を作る際に必要な枝である．r に e_l の ID を保存する．処理 5 を図 11 に示す．interlace 定理を利用するために必要な行列を作成するのが処理 7 である．接続行列で interlace 定理を利用すれば，部分グラフのフィルタリングができる．グラフ G'_l, G'_s の接続行列 g'_l , g'_s を作る手順を図 12，13，14，15，16 に示す．図 13，14 は，枝の接続がない頂点を除くのに加え，行列の行と列の番号を得る為の処理である．行列 g'_l, g'_s の固有値 α,θ を計算する．本稿では，MATLAB を使用して固有値を求める．固有値を使って，interlace 定理によるフィルタリングを行う．interlace 定理を満たさなければ部分グラフではないと判定する．手順を図 17 に示す．interlace 定理によってフィルタリングが出来なかった場合，VF2 アルゴリズム [2] を用いて部分グラフ同型判定を行う．

5. 評 価 実 験

5.1 速 度 実 験

G_s が G_l の部分グラフであるかの判定を行う処理を 1000 回行った．500 回は部分グラフ同型であるものを判定し，500 回は，部分グラフ同型ではないものを判定した．VF2 のみで部分グラフ同型判定を行う場合と提案手法（図 9）で行う場合について，判定が終了するまでに掛かる計算時間の測定を行った．グラフは以下のものを使用した．

- 頂点数 80，平均枝数 160，頂点のラベルに 1 から 4 までの数値を使用したグラフ G_s

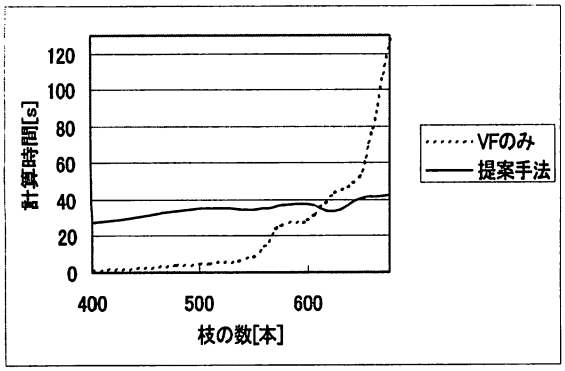

図 18　部分グラフを判定する処理時間

表 1　フィルタリングされた数と部分グラフの数

判定方法	ラベル	Interlace	部分グラフ同型	計算時間
VF のみ	-	-	0	386.0[sec]
提案手法	817	146	0	65.8[sec]

- 頂点数 100，頂点のラベルに 1 から 8 までの数値を使用したグラフ G_l

G_s に使用したラベルは，G_l に使用したラベルより少なくした．G_l の枝数を 300 から 700 まで変化させた場合における VF2 のみの計算時間と提案手法による計算時間を図 18 に示す．G_l の枝数が 600 以上になると，VF2 のみで部分グラフ同型判定を行うと時間がかかるようになる．これは，G_l の枝の数が増えた結果，VF2 が調べなければならないグラフの組み合わせが増えたためであると考えられる．部分グラフの判定について，G_s に使用されるラベルが G_l に使用されるラベルより少なく，グラフ G_l の枝数が多い場合，提案手法が有効である事が確認できた．頂点，枝の両方にラベルを使用すると，VF2 のみで部分グラフ同型判定を行う方が計算時間が速い．また，頂点のみにラベルを使用する場合，使用するラベルが多くなると，VF2 のみで部分グラフ同型判定を行う方が計算時間が速くなる．

5.2 フィルタリング実験

G_s が G_l の部分グラフであるかの判定を行う処理を 1000 回行った．G_l, G_s はそれぞれ，以下のものを使用した．

- 頂点数 80，平均枝数 150，頂点のラベルに 1 から 6 までの数値を使用したグラフ G_s
- 頂点数 100，平均枝数 650，頂点のラベルに 1 から 8 までの数値を使用したグラフ G_l

VF2 のみで部分グラフ同型判定を行った場合について，部分グラフ同型の数，計算時間を表 1 に示す．提案手法については，ラベルによってフィルタリングされた数，interlace 定理によってフィルタリングされた数も示す．フィルタリングされたグラフは部分グラフ同型ではないと判定する．

VF2 のみで部分グラフ同型を 1000 回行った場合，部分グラフ同型であるものが 0 個であると判定されるまでに 386.0[sec] 掛かった．提案手法を用いた場合，ラベルによるフィルタリング $\xi(e_s) = \xi(e_l)$ によって 817 個がフィルタリングされ，interlace 定理によって 146 個がフィルタリングされた．残りの 37 個は VF2 により部分グラフ同型判定を行ったが，全体の計算時間は 65.8[sec] となった．提案手法が VF2 と比べて 320[sec] ほど速く部分グラフ同型判定を行う事ができた．ラベル，interlace 定理によるフィルタリングを前処理として利用し，部分グラフ同型判定を行う提案手法が，組み合わせ的に部分グラフ同型判定を行うよりも有効であることが分かった．

6. おわりに

本稿では，部分グラフ同型判定の前処理として，ラベルを利用してグラフをフィルタリングすると共に，interlace 定理を利用するために必要な行列のサイズを縮小する手法を提案した．以下の条件を満たした場合において，部分グラフ同型判定を行う場合，組み合わせ的に部分グラフ同型判定を行うよりも，提案手法を用いてグラフをフィルタリングして部分グラフ同型判定を行う方が有効である事を確認した．

- G_s に使用されるラベルが G_l に使用されるラベルより少ない
- $|V_l|$, $|V_s|$ の差が少ない
- $|E_s|$ より $|E_l|$ が大きい

今後の課題として，多様なデータによる実験，サイズを縮小した行列とサイズを縮小しない行列による interlace 定理の判定精度の変化を検証等が挙げられる．

謝辞　本研究の一部は，(独) 日本学術振興会科学研究費補助金基盤研究 (C)(課題番号:19500089) による．

文　献

[1] J.R. Ullmann, "An algorithm for subgraph isomorphism," JAssoc. Comput. Mach, vol.23, pp.31-42,1976.

[2] L.P. Cordella, P. Foggia, C. Sansone, M. Vento, "An Improved Algorithm for Matching Large Graphs," Proc. of the 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition, Ischia, May 23-25, pp. 149-159, 2001.

[3] B.T. Messmer and H. Bunke, "Efficient subgraph isomorphism detection: A decomposition approach," IEEE Trans. Knouledge And Data Engineering, vol.12, pp.307-323, 2000.

[4] X. Yan, P. S. Yu, J. Han, "Graph Indexing: A Frequent Structure-based Approach," Proc. 2004 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'04), Paris, France, June 2004.

[5] K. Yoshida and H. Motoda, "Concept Learning from Inference Patterns", Artificial Intelligence, Vol. 75, No.1, pp. 63-92, (1995).

[6] Willem H. Haemers, "Interlacing Eigenvalues and Graphs," Linear Algebra Appl. 226, pp.593-616, 1995.

[7] X. Yan, P. S. Yu, J. Han, "Substructure Similarity Search in Graph Databases," Proc. of 2005 Int. Conf. on Management of Data (SIGMOD'05), 2005.

[8] 高橋由雅，藤島悟志，加藤博明，"化学物質の構造類似性に基づくデータマイニング，" J.Comput. Chem.Jpn.,vol.2, No.4,pp.119-126, 2003.

[9] 長屋未来，片山　薫，石川　博，"大規模グラフを対象とした部分グラフ同型判定における Interlace 定理の利用，"電子情報通信学会第 17 回データ工学ワークショップ DEWS2006,2006.3

[10] 下野弘貴，佐藤正実，北澤仁志，"特徴空間中の部分グラフ間距離の高速計算による実時間行動識別，"信学技報，vol. 104, no. 669, PRMU2004-197, pp. 109-114, 2005 年 2 月．