グラフの連結性に基づく Messmer らの部分グラフ同型判定手法の改良

Improvement of Messmer's Approach to Subgraph Isomorphism Detection based on Connectedness of Graphs

西村 将太郎♥ 片山 薫♦
太田 学♠ 石川 博♦

Shotaro NISHIMURA Kaoru KATAYAMA
Manabu OHTA Hiroshi ISHIKAWA

二つのグラフが与えられた時，一方のグラフがもう一方のグラフに含まれるかどうかを判定する問題を部分グラフ同型判定問題という．この問題は NP 完全なので，多くのグラフを扱う際には膨大な計算コストが必要である．Messmer らはグラフを，グラフの分解によって得られる特別なデータ構造に変換後，部分グラフ同型判定を行うことによって，この問題を効率的に解くアルゴリズムを提案している．我々はグラフ分解において必ず連結グラフが生成されるようにすることで，より効率的に部分グラフ同型判定を行う手法を提案する．

The subgraph isomorphism problem is the problem of whether one graph is contained in another graph, given two graphs. An enormous calculation cost is necessary for this problem when large quantities of graphs are handled because it is NP complete. Messmer et al. propose the algorithm to solve this problem efficiently based on graph decomposition. We improve the Messmer's approach so that connected graphs are formed in the decomposition of each graph.

1. はじめに

二つのグラフが与えられた時，一方のグラフがもう一方のグラフに含まれるかどうかを判定する問題を部分グラフ同型判定問題という．部分グラフ同型判定は，パターン認識やコンピュータビジョンなどの情報学の分野の他，化学や生物学などの様々な分野において，広く応用されグラフマッチングとも呼ばれる．しかし，この問題は NP 完全なので，多くのグラフを扱う際には膨大な計算コストが必要である．

関連研究としては，検索範囲を著しく減少させる手続きである，バックトラックに基づくアルゴリズムが Ullmann [2] によって提案されている．グラフ同型性と，部分グラフ同型判定の両方のためのアルゴリズムであり，今日まだ正確なグラフマッチングとして一般に使われるものの 1 つである．さらに，グラフをカノニカルなフォームに変形してから同型を判定する Nauty algorithm[3]がある．VF[4]は，グラフ同型，およびサブグラフ同型判定手法で，複雑で大きなグラフを扱うことができる．

Messmer ら[1]はグラフ集合を，グラフの分解によって得られる特別なデータ構造に変換後，部分グラフ同型判定を行うことによって，この問題を効率的に解くアルゴリズムを提案している．この論文では，Messmer らのアルゴリズムを元にした，より効率的な部分グラフ同型判定の手法を提案する．グラフ分解を利用したアプローチは，例えば画像データデータベース中の各画像が，必要な複数の特徴を持っているかを調べるといったような，多くのデータに対して同じ問い合わせを繰り返す必要のある処理に有効である．

2. 提案手法の概要

2.1 Messmer らの部分グラフ同型判定手法

グラフの集合 $G_1,\ldots,G_n$ が与えられた時，それらがグラフ G_I に含まれるかどうかを判定する問題を考える．本稿では，$G_1,\ldots,G_n$ をモデルグラフ，G_I を入力グラフと呼ぶことにする．単純な方法としては，例えば Ullman のアルゴリズムを利用して各モデルグラフに順次入力グラフをマッチする等が考えられるが，この問題は NP 完全であり，大量のグラフを扱う際には膨大な計算コストが必要である．

Messmer らはこの問題を解くため，以下のようなアルゴリズムを考案した．入力グラフを個々に各モデルグラフとマッチングする代わりに，まずモデルグラフ G_1，G_2 を，図 1 に示すように分解する．あるグラフ G_i（又は S_i）を分解する際，それ以前の分解によって生成されたグラフの中に，グラフ G_i に含まれるグラフ（部分グラフ）があれば，G_i をその部分グラフとそれ以外の部分に分解する．そのような部分グラフがなかったらランダムに分解する．図 1 では，G_1 がランダムに分解され S_1 と S_2 が生成される．更に S_1 は分解され S_5 と S_6 が生成される．G_2 の分解では，G_2 の部分グラフ S_3 が発見され，G_2 は S_3 と G_2 から S_3 を引いたグラフ S_4 が生成される．それから，入力グラフ G_I と部分グラフをマッチングして，最終的に完全なモデルグラフとの部分グラフ同型を検出する．この方法による利点は，分解によって得られた異なるグラフに複数回現れる部分グラフ(例えば図 1 の S_3)が，入力グラフとマッチングされるのは一度だけであるということである．

この方法は 2 つのパートからなる．まずモデルグラフが分解されるプロセスで，得られた部分グラフは図 1 のような特別なデータ構造で表される．次のプロセスで，前プロセスで生成されたデータ構造で表されたモデルグラフと入力グラフをマッチングする．

Messmer らの手法の欠点は，モデルグラフを分解する際に，枝が切られ，連結ではない部分グラフが生成されることである．それによって計算コストが非常に増加してしまう．詳しくは次の 2.2 章で述べる．

2.2 グラフ分解方法の改良点

Messmer らのアルゴリズムは，グラフの分割において切られる枝について考慮していないので，たくさんの枝が切られたり，連結ではない部分グラフが生成される可能性がある．

♥ 学生会員 東京都立大学大学院工学研究科修士課程
nishi-no-heya_2@msj.biglobe.ne.jp
♦ 正会員 東京都立大学大学院工学研究科
{katayama, ishikawa}@eei.metro-u.ac.jp
♠ 正会員 岡山大学大学院自然科学研究科
ohta@suri.it.okayama-u.ac.jp

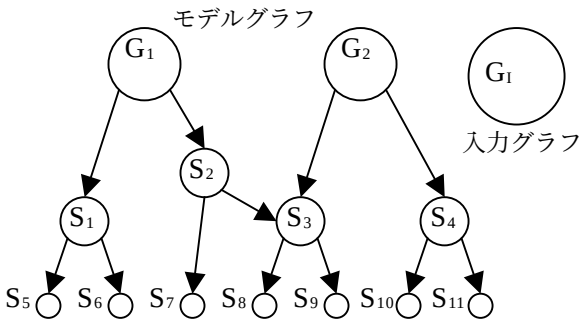
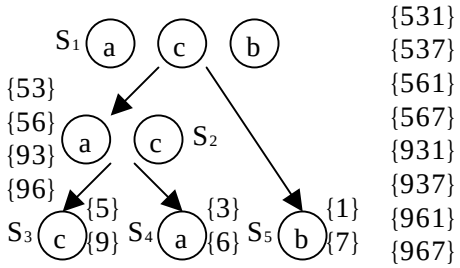


図 1　Messmer らの方法
Fig.1　Messmer's approach



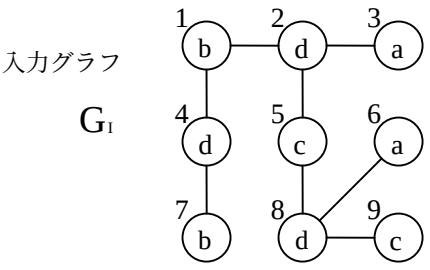


図 2　枝がないモデルグラフに対する部分グラフ同型判定
Fig.2　Subgraph isomorphism for model graph with no edge

このような部分グラフが生成されると，部分グラフ同型判定に要する計算コストが非常に増加してしまう．特に疎なグラフは枝が少ないので，このような現象が起きやすい．

例を図 2 に示す．丸の中のアルファベットは頂点のラベルで，入力グラフ G_I の頂点の横にある数字は頂点の ID である．分解されたモデルグラフの部分グラフ S_1,…, S_5 の横の{ }で囲まれた数字は，割り当てられた入力グラフ G_I の頂点の ID である．上の図は，枝を持たない 3 つの頂点だけからなるグラフ S_1 の分解例である．まず S_3 は頂点のラベルが c なので，G_I においてラベルが c である ID が 5,9 の頂点が割り当てられる．同様に S_4 は頂点のラベルが a なので G_I の ID3,6 の頂点が割り当てられ，S_5 は頂点のラベルが b なので G_I の ID1,7 の頂点が割り当てられる．S_2 は枝がないので，G_I において割り当てられた頂点の間にも枝がなければ必ず部分グラフである．G_I の ID5,9 の頂点と ID3,6 の頂点の間には枝がない．よって S_2 は 4 通りの割り当てができる．さらに，S_1 には枝がなく，G_I の ID5,3,6,9 の頂点と ID1,7 の頂点のそれぞれの間にも枝がないので 8 通りの割り当てが生成される．実際は，より頂点の数，モデルグラフの個数が多いので，非常に多くの割り当てが生成されてしまうことがある．

グラフ G を分割する際，Messmer らの方法では(1)今までの分解で生成されたグラフの中で最大の部分グラフ S_{max} と，$G-S_{max}$（G と S_{max} の差異）に分割，(2)ランダムに分割，の二つの分け方がある．我々の方法では，この両方の場合において，分解された部分グラフが必ず連結グラフになるようにする．(1)では，$G-S_{max}$ が，連結グラフになるような S_{max} をとり，(2)においては，Kernighan/Lin アルゴリズムに，分解されたグラフが必ず連結グラフになるという条件を加えた分解方法で，切られる枝が少なく，かつ分解されたグラフが連結グラフになるようにする．

3. 提案手法

以下では，例を用いて提案手法を説明する．

3.1 モデルグラフ分解

図 3 にモデルグラフ G1，G2 の分解と入力グラフ GI を示す．ここでは枝のラベルは省略する．モデルグラフは G1 と G2 である．グラフの頂点の中のアルファベットは頂点のラベル，モデルグラフの G1 と入力グラフ GI の頂点の横の数字は，説明のために用いる頂点の ID である．グラフの横の D_i は，G_i の分解によって生成されたデータ(分解データ)で、$D_i=(G_i,S_1,S_2)$で定義される．ここで，S_1，S_2 は G_i の分解によって生成されたグラフである．D_i の i は，D_i が分解データ集合 D に入れられる順を示す．

はじめは D=φなので G1 は Kernighan/Lin アルゴリズムで分解される．例えば，（実際にはすべての可能な分解について考えるが）G1 を(1) S_1={1,2,3}, S_2={4,5}，(2) S_1={1,4}, S_2={2,3,5}，(3) S_1={3.5}，S_2={1,2,4}の 3 つの場合に分解することについてだけを考える（{ } の中の数字は G1 の頂点の ID）．まず，Kernighan/Lin アルゴリズムによって，G1 を頂点の数が同数（奇数個ある時は，k 個と k+1 個）になるよう分解し，かつ切られる枝の個数を最小にするような場合を選択する．切られる枝の個数は，(1)は 3 個，(2)と(3)は 2 個なので(2)，(3)を選択する．次に，(2)は連結グラフではないので(3)のように分割し，図のようになる．

さらに G1 は分解されて，頂点 1 つになったとき初めて D_i が D に入れられる．ある分解データ D_i の G_i を頂点一つになるまで分解することによって生成されたすべての分解データが D に入れられてから，その分解データ D_i 自身が D に入れられる（例えば，D_4，D_5，D_6，D_7 が D に入れられてから D_8 が入れられる）．

G2 の分解を考える．まず，G2 の S_{max} を探す（G1 を分解している時も，D≠φの時点から S_{max} を探している）．探す順番は D_1，D_2，D_3,…である．しかし，頂点が 1 つから成るグラフは S_{max} とはみなさない．G_3 は部分グラフではない．G_8 は部分グラフであるが，$G2-G_8$ が連結グラフではないので S_{max} ではない． G_7 は部分グラフで，$G2-G_7$ が連結グラフなので S_{max} である．G_9 は部分グラフではない．G2 を分解する時点で D に格納されている分解データ D_i は D_9 までなので，S_{max} は D_7 の G_7 となり，G2 は G_7 と $G2-G_7$ に分割される．G_{14} は分解データ集合 D に S_{max} がないので，Kernighan/Lin アルゴリズムによって分解される．G2 はさらに分割され，図 3 のようなデータが生成される．

3.2 部分グラフ同型判定

図 3 を用いて例証する．分解されたそれぞれのグラフの横の{ }内に，割り当てられた頂点が示されている．頂点が割り当てられない場合，又はそれまでの計算から判定可能で計算不要の場合は“ dead ”が示されている．

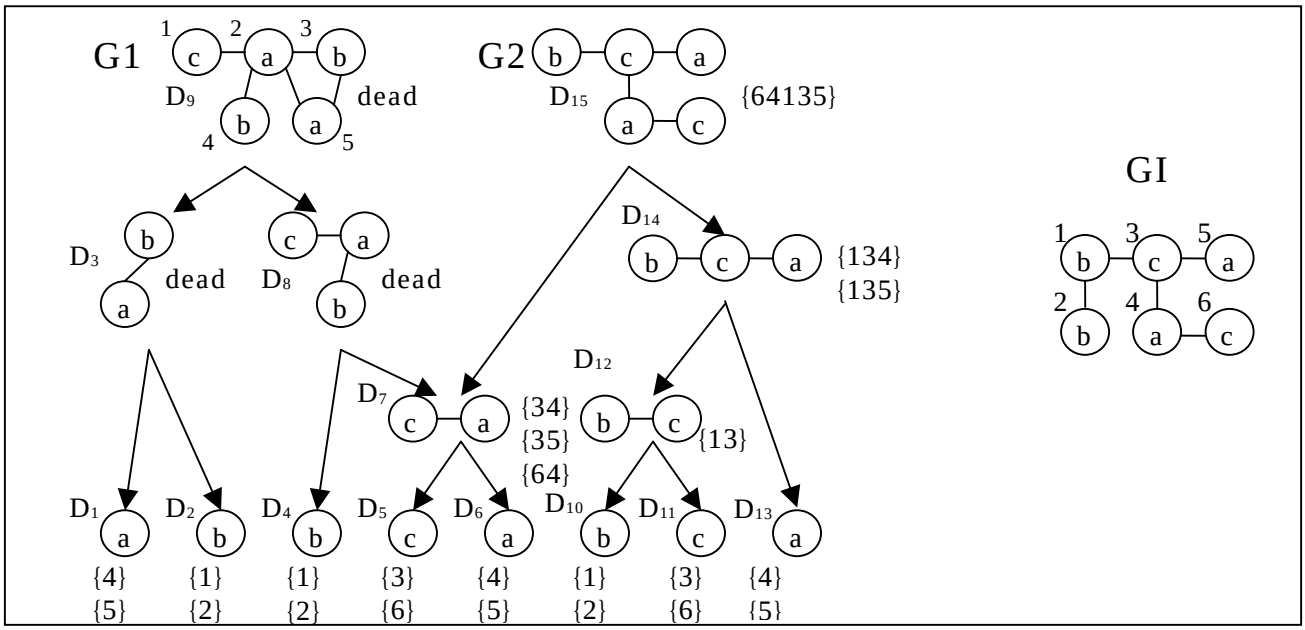


図 3　モデルグラフの分解と部分グラフ同型判定
Fig.3　Decomposition of model graphs and the subgraph isomorphism detection

D_i の i=0,1,2…の順に部分グラフ同型判定をする．G_1 のラベルは a なので，入力グラフのラベルが a である頂点{4},{5}が割り当てられる．G_2 も同様にして頂点{1},{2}が割り当てられる．G_3 は入力グラフの部分グラフではないので“dead”である．ここで G_3 を含んでいる G_9 も必ず部分グラフにはならないので“dead”である．

次に G_4，G_5，G_6 にそれぞれ{1},{2}と{3},{6}と{4},{5}を割り当てる．G_7 は G_5 と G_6 に割り当てられた頂点を結合して部分グラフになるものを選ぶ．G_7 の頂点の間には枝があり，GI の ID3 と ID4，ID3 と ID5，ID6 と ID4 の頂点間にも枝があるので，G_7 には{34}(G_5 の{3}と G_6 の{4}を結合しそのまま並べて示す)，{35}，{64}が割り当てられる．G_8 は G_4 のラベル b の頂点と G_7 のラベル a の頂点の間に枝があるが GI にはないので部分グラフではなく“dead”になる．G_9 は“dead”であるので計算不要である．同様にして G_{10} から G_{14} まで計算する．G_{15} は G_7 と G_{14} を結合する．G_7 と G_{14} の間で同じ頂点 ID が割り当てられていない組み合わせは{64}と{135}だけである．G_{15} において G_7 と G_{14} の間には，ID4 と 3 が割り当てられた頂点の間にだけ枝がある．GI においても，{64}と{135}の間にある枝は ID3 と 4 の頂点の間だけである．よって，G2 に{64135}が割り当てられ，G2 は入力グラフ GI の部分グラフであることがわかる．

4. 性能評価実験

Messmer らのアルゴリズムと我々のアルゴリズムを比較した．一つのグラフの平均頂点数，50 個のときのモデルグラフの個数を変化させたときに，モデルグラフの分解，および部分グラフ同型判定をするのに要した処理時間を測定した．

4.1 実験環境

Pentium(R)4 プロセッサ 3GHz，メモリ 1GB，OS として Windows XP を搭載した PC を使用し， Visual C++を使用して開発を行った．実験で使用するグラフデータは，蔵持らの開発したグラフ生成ソフトウェアを利用した．

4.2 実験結果

図 4，5 は，グラフ集合に含まれる一つのグラフの平均頂点数と，それらをグラフ分解，部分グラフ同型判定をするのに要する時間との関係を示している．

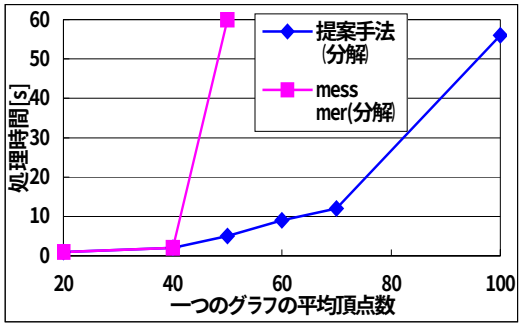


図 4　平均頂点数とグラフ分解に要する時間
Fig.4　Average number of vertices of graphs and processing time for their decomposition

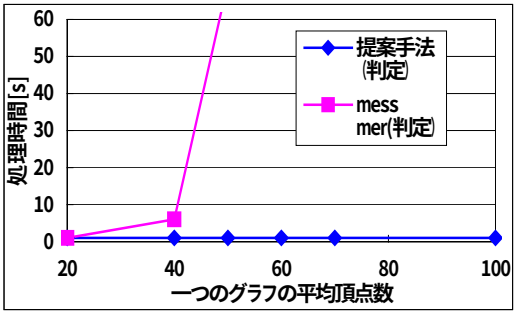


図 5　平均頂点数と部分グラフ同型判定に要する時間
Fig.5　Average number of vertices of graphs and processing time for the subgraph isomorphism detection

提案手法は一つのグラフの平均頂点数が増えるにしたがって，処理時間もなだらかに増加するが，Messmer らの方法では一つのグラフの平均頂点数が 50 個から急激に増え，50 個以上では測定不可能（使用可能なメモリ量を超えたため）となった．Messmer らの手法では，一つのグラフの平均頂点数が 50 個以上では図 2 のような枝がない分解が生成され，急激に処理時間が増加したと考えられる．メモリオーバーとなったのは，割り当てられた頂点のデータが大きすぎたためである．

図 6, 7 は一つのグラフの平均頂点数が 50 個におけるモデルグラフの数と，それをグラフ分解，部分グラフ同型判定するのに要する時間との関係を示している．

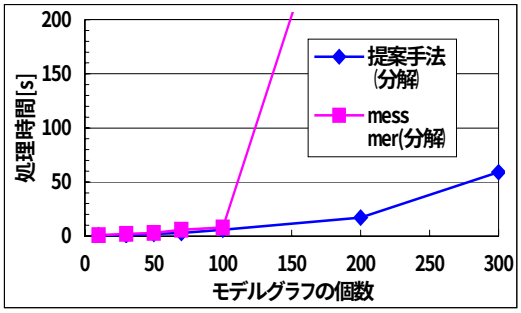


図 6　平均頂点数が 50 個におけるモデルグラフの個数と，モデルグラフの分解に要する処理時間

Fig.6　Number of model graphs whose average number of vertices is 50 and processing time for their decomposition

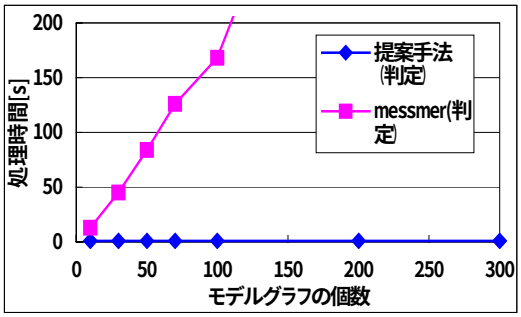


図 7　平均頂点数が 50 個におけるモデルグラフの個数と，部分グラフ同型判定に要する処理時間

Fig.7　Number of model graphs whose average number of vertices is 50 and processing time for the subgraph isomorphism detection

図 6，7 においてモデルグラフの個数が 100 を越えると測定不可能（使用可能なメモリ量を超えたため）であった．提案手法は，平均頂点数が増えても処理時間が急激に増加することはなく処理を行った．

これらの実験結果から，提案手法により，Messmer らの手法において枝がない分解が生成され，頂点の割り当てが急増する問題を解決し，より大規模なグラフを扱えるようになったことが分かる．

5. まとめと今後の課題

Messmer らのアルゴリズムを元にした，より効率的な部分グラフ同型判定の手法を提案した．グラフを分解する時必ず連結グラフになるようにすることによって，Messmer らの手法における問題を解決した．頂点数が多いグラフにおいて我々の手法が有利であることが示された．

今後の課題は，代表的な部分グラフ同型判定アルゴリズム VF と我々の提案手法を比較することである．更に，部分グラフ同型判定のステップにおいて，入力グラフがモデルグラフに含まれるかという問題を扱えるよう拡張することを考えている．

[謝辞]

実験用グラフデータ生成ソフトウェアを提供頂いたミネソタ大学蔵持道広氏に感謝致します．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(B)(2)(課題番号:16300030)による．

[文献]

[1] Bruno T. Messmer and Horst Bunke, “Efficient Subgraph Isomorphism Detection: A Decomposition Approach” IEEE Transactions on Knowledge and Data Engineering, 12(2), 2000.

[2] J.R. Ullmann, “An Algorithm for Subgraph Isomorphithms”, Journal of Association for Computing Machinery, vol.23, pp.31-42, 1976.

[3] B.D. Mckay, “Practical Graph Isomorphism,” Congressus Numerantium, vol. 30, pp. 45-87, 1981.

[4] Luigi.P.Cordella, Pasquale. Foggia, Carlo. Sansone, Mario. Vento, “A (sub)Graph Isomorphism Algorithm for Matching Large Graphs”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.26, no.10, pp.1367-1372, October 2004.

西村 将太郎　Shotaro NISHIMURA
東京都立大学大学院電気工学研究科修士課程在学中．日本データベース学会学生会員．

片山 薫　Kaoru KATAYAMA
東京都立大学大学院工学研究科助手．2000 年京都大学大学院情報学研究科社会情報学専攻博士後期課程了．博士(情報学)．データベースシステムに関する研究開発に従事．情報処理学会，日本データベース学会各会員．

太田 学　Manabu OHTA
岡山大学大学院自然科学研究科助教授．東京都立大学大学院工学研究科助手を経て 2005 年より現職．1999 年東京大学大学院工学系研究科電気工学専攻博士課程了．博士(工学)．情報検索，データマイニングとその Web への応用に興味を持つ．情報処理学会，電子情報通信学会，日本データベース学会，IEEE 各会員．

石川 博　Hiroshi ISHIKAWA
東京都立大学大学院工学研究科教授．富士通研究所を経て 2000 年より現職．1979 年東大・理卒．博士（理学）．データベースシステムの研究開発に従事．情報処理学会，ACM，IEEE 各会員．International Journal on Very Large Data Bases Editorial Board，ACM Sigmod Japan 評議員，日本データベース学会理事．情報処理学会データベースシステム研究会主査．情報処理学会論文誌(データベース)編集委員長．ナント大学(フランス)エコールポリテクニック招聘教授．著書に「E－ビジネス技術入門書」「次世代データベースとデータマイニング」(CQ 出版)など．