様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　５ 年　 ６ 月　 ７ 日現在

機関番号：３４５０４
研究種目：基盤研究(C)（一般）
研究期間：2020～2022
課題番号：２０Ｋ１１８３５

研究課題名（和文）グラフデータベースから類似グラフを検索する計算フレームワークの構築

研究課題名（英文）Construction of a framework for searching similar graphs from graph databases

研究代表者
猪口　明博（Inokuchi, Akihiro）

関西学院大学・工学部・教授

研究者番号：７０４５２４５６

交付決定額（研究期間全体）：（直接経費）　　3,300,000円

研究成果の概要（和文）：本研究の成果の１つ目は，グラフ検索のためのフレームワークをデザインしたことである．これは，包摂グラフ検索，類似包摂グラフ検索，部分グラフ検索などのグラフ検索問題を解くための共通のソフトウェア基盤となっている．２つ目の成果は，類似包摂グラフ検索問題を新たに定義し，それを解くための手法を提案したことである．類似包摂グラフ検索問題とは，多数のグラフからなるデータベースとクエリグラフqが与えられたときに，qの部分グラフに類似するグラフをデータベースから取得する問題である．部分グラフを得ること，グラフの類似性を計算することはともにNP完全ではあるが，高速に動作する類似包摂グラフ検索を実現した．

研究成果の学術的意義や社会的意義
グラフ形式で蓄積された大量のデータを利活用するには，所望のデータを素早く得るための検索技術が必要となる．部分グラフ同型判定問題はNP完全であるので，大量のグラフデータに対して，２つのグラフの間の部分グラフ同型判定問題を複数回解くことは適切ではない．本研究で実現した検索技術は，そのような問題を解かず，データベースの複数のグラフとクエリグラフの間のグラフ同型判定問題を同時に解くことができる．また，部分グラフ検索，包摂グラフ検索，類似包摂グラフ検索に対応できるようにソフトウェアをデザインしており，様々な検索問題を解くために容易に拡張できる．

研究成果の概要（英文）：The first contribution of this study is to design the framework for graph search. It is used as our common software foundation for solving graph search problems such as supergraph search, similar supergraph search, and subgraph search. The second one is that we newly defined the similar supergraph search problem and proposed the method for solving it. The similar supergraph search problem is the problem of obtaining graphs similar to subgraphs of a query graph q from a database of many graphs. Although both obtaining subgraphs and computing graph similarity are NP-complete, we achieved a search method that works fast.

研究分野：データベース

キーワード：グラフ　グラフ検索　類似グラフ　グラフ編集距離　部分グラフ同型判定

様　式　C－１９、F－１９－１、Z－１９（共通）

1. 研究開始当初の背景

グラフは物事の関係を表現するのに有用なデータ構造である．例えば，1 つの分子内の原子や化学結合をそれぞれグラフの頂点や辺に対応させると，1 つの化合物は 1 つのグラフで表現することができる．また，人間や人間関係をそれぞれグラフの頂点や辺に対応させると，人間関係もグラフで表現することができる．それ以外にも，グラフで表現できる実世界の対象物は様々存在する．このためグラフ形式で蓄積されるデータは非常に多く，急速な勢いで増加している．

このような背景から，グラフ形式で蓄積されたデータを利活用することは重要であり，その活用方法の 1 つは，蓄積されたデータの中から所望のグラフを検索することである．グラフ検索の代表例としては，n 個のグラフ{g1, g2, … ,gn}が蓄積されているデータベース D にクエリグラフ q を与え，q の条件に合致するグラフを D から発見することである．これまで，以下の表に示す部分グラフ検索，包摂グラフ検索，類似グラフ検索などの問題が解かれてきた．ここで，g が g'の部分グラフであることを g⊆g'で表す．

表 1 グラフ検索問題の種類

	問題名	定式化	既存手法
①	部分グラフ検索	{gi | q⊆gi} ={gi | ∃s⊆gi *s.t.* s=q}	GraphGrep, Grapes, SING, CT-Index, vcFV GDIndex, GCode, SwiftIndex, Tree+Delta, CP-Index, gIndex, FG-Index, Lindex
②	包摂グラフ検索	{gi | gi⊆q} ={gi | ∃s⊆q *s.t.* s=gi}	cIndex, GPTree, PrefIndex, LW-Index, CodeTree, DGTree
③	類似グラフ検索	{gi | sim(q,gi)≦ θ }	ML-index, GDDA
④	類似部分グラフ検索	{gi | ∃s⊆gi *s.t.* sim(s,q)≦ θ }	未確立（研究開始時点）
⑤	類似包摂グラフ検索	{gi | ∃s⊆q *s.t.* sim(s,gi)≦ θ }	未確立（研究開始時点）

g が g'の部分グラフであるかを判定する問題は部分グラフ同型判定と呼ばれる．部分グラフ同型判定問題は NP 完全であることが知られている．このため，D のあるグラフと q が，部分グラフの関係にあるかを逐次的に調べることは効率が悪い．そこで，部分グラフ検索や包摂グラフ検索問題を解く手法の多くは，以下の 3 つの手順を取るものが多い．(1) q を受け取る以前に D に対して索引を構築する．(2) q を受け取ったら，索引を巡回し，解とならないものを除去する．(3)除去できなかったグラフと q の間の部分グラフ同型判定問題を解く．この手順を filtering & verification と呼ぶが，上記の表の既存手法の多くは，この手順に基づいている．

類似グラフ検索とは，q に類似するグラフを D から検索する問題である．2 つのグラフ g と g'の類似度を sim(g,g')で表す．グラフの類似度を測る尺度は様々あるが，その代表例は，グラフ編集距離であり，グラフ編集距離を求める問題もまた NP 完全である．

このような背景の中，研究代表者は，包摂グラフ検索問題を解く手法として，CodeTree と呼ぶ手法を開発していた．CodeTree では，グラフはグラフコードで表現される．D の各グラフをグラフコードで表現し，それらのコードの接頭辞木を索引とした．クエリグラフを受け取ったら，その接頭辞木を巡回し，解を探索する．CodeTree の特徴は，それまでの手法とは異なり，filtering と verification を区別せず，filtering をしながら同時に verification を行う．CodeTree は，q と D の複数のグラフとの間の部分グラフ同型判定問題を同時に解くことができるために，高速に動作する．

2. 研究の目的

本研究では以下のように目的を定めた．

(1) グラフ検索の計算フレームワークの構築

上記の表に示すように，①から③の示す問題を個別に解く様々な手法が存在する．しかし，部分グラフ検索問題と包摂グラフ検索問題の両問題のように複数の問題に適用できる手法は存在していなかった．しかし，①から⑤の問題は同じグラフ検索を扱うための部分的な処理に共通点があるはずであり，その共通点を API として整備し，その上に個別の問題を解くようにソフトウェアをデザインしたほうが，ソフトウェアの汎用性が高く，その保守性も高いと考えられる．そこで，グラフ検索の計算フレームワークの構築を目指した．

(2) 類似するグラフを検索する手法を確立

上記の表に示すように，本研究開始時点では，類似部分グラフ検索問題④や類似包摂グラフ検索問題⑤は定義されておらず，当然のことながらそれらの問題を解く手法も存在していなかった．そこで問題④や⑤を解く手法を確立することを目標とした．

3. 研究の方法

まず，研究の目的で述べた「グラフ検索の計算フレームワークの構築」について述べる．本研究では CodeTree をベースとする．前述のように，CodeTree は，グラフをグラフコードで表す．グラフコードとして，AcGM コードや DFS コードが知られている．CodeTree の基本的な考え方は様々なグラフコードに適用できるという意味において CodeTree は汎用性があり，それが CodeTree 特徴でもある．そのアイデア自体は，本研究（3 年間）の準備段階においてできていたが，それを実現するソフトウェアが未完成であった．具体的には AcGM コードに基づく CodeTree と DFS コードに基づく CodeTree が別のソフトウェアとして作られており，ソフトウェアの保守や評価実験に大きなコストがかかっていた．2 つのソフトウェアを共通化するために，Java の Interface を活用した．V 個の頂点からなるグラフは，AcGM コードで V 個のフラグメントの系列で表される．また，E 個の辺からなるグラフは，DFS コードで E 個のフラグメントの系列で表される．このような様々な共通点を見出し，我々の CodeTree のソフトウェアでは，CodeFragment や GraphCode と呼ばれる Interface をデザインした．そのうえで，AcGM コードや DFS コードはそれらの実装クラスとして定義される．また，接頭辞木を巡回するメソッドは，特定のグラフコードには依存せず，上記の Interface のみを介して，定義されている．これを実現するために，巡回途中で保持すべき情報を保持する Interface として SearchInfo をデザインした．木の巡回は特定のグラフコードに依存せずプログラミングされているために，AcGM コードや DFS コード以外のコードを新たに定義した時には，このソフトウェアに容易に組み込めるようになっている．実際に，拡張 AcGM コードと呼ばれるグラフコードを新たに提案したが，AcGM コードの拡張クラスとして，僅か数行程度書き加えるだけで，CodeTree のソフトウェアが動作した．

このソフトウェア構築のあとに，類似包摂グラフ検索の手法，部分グラフ検索，包摂グラフ検索の並列計算の手法を順次作成していったが，このフレームワークのおかげで，個々のソフトウェアを作成する時間を大きく削減できた．

続いて，研究の目的で述べた「類似するグラフを検索する手法を確立」について述べる．それを述べるにあたり，この研究の必要性について述べる．2020 年以降，コロナウイルスの感染が広がり，社会問題となった．その感染症の有望な抗ウイルス剤として，ファビピラビル（医薬品名：アビガン）が検討された（ただし，結果的には，コロナ感染症の医薬品とはならなかった）．ファビピラビルは元々抗インフルエンザウイルス剤として開発されたものである．ファビピラビルがリボース化されると，その構造はアカデシンに似る．アカデシンは RNA の材料となるグアノシンやアデノシンの前段階のイノシンに至る前駆物質である[3]．そのため，ウイルスの RNA 依存性 RNA 合成酵素は，伸長中のウイルス RNA に，ファビピラビルをグアノシンなどと間違えて取り込んでしまい，そこで，RNA 合成が停止する．これにより，体内でのウイルスの増殖を抑制することができる．図 1 はグアノシン，イノシン，アカデシン，リボース化されたファビピラビルの化学構造である[5]．赤で描かれた部分構造αは一致しているので，新薬のシード化合物探索の際には，多数の化合物を含むデータベースに対して，部分構造αをクエリとして検索する方法（部分構造検索）が考えられる．一方，新薬開発に重要な部分構造が事前に多数既知であれば，このような部分構造をデータベースに蓄積しておき，ある新規化合物が開発された際には，それをクエリとして後者のデータベースを検索すること（包摂構造検索）が考えられる．類似包摂グラフ検索とは，後者に関連する手法である．図 1 の青で描かれた部分構造は，部分構造αに類似しているが，完全には一致していない．抗インフルエンザ剤に寄与する部分構造αがデータベースに登録されている状況下で，リボース化されたファビピラビルをクエリとして与え，部分構造αが検索結果として見つかれば有用であると考えられる．そのような検索が高速に実現できる方法論を目指した．

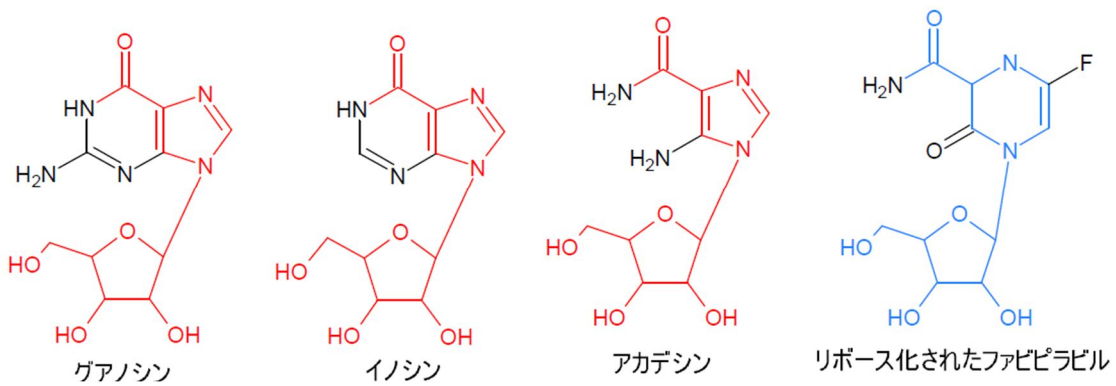

図 1 類似した化合物

類似包摂グラフ検索問題は以下のように定義される．

$$S = \{g_i \in G \mid \exists q' \subseteq q \; s.t. \; ed(g_i, q') \leq \theta\} \qquad (1)$$

ここで ed はグラフ gi と q’ (q’はクエリグラフ q の部分グラフ)のグラフ編集距離を求める関数であり，それが閾値θ以下のグラフをデータベースから検索する．

前述のように，部分グラフ同型判定問題，及びグラフ編集距離を計算する問題はともに NP 完全である．このため，単純な方法で式(1)を満たすグラフを探索すると計算爆発を起こす．そこで，コードフラグメントに対して，編集距離を求めることができることを示し，接頭辞木の巡回において節点を 1 つ辿るごとに，閾値θを超えていないかをチェックする．閾値を超えた場合には，その節点の子節点の探索を枝刈りし，バックトラックすることで，効率的な巡回を実現した，

4．研究成果

学会論文等で公開した研究成果のうち代表的なものを抜粋して示す．

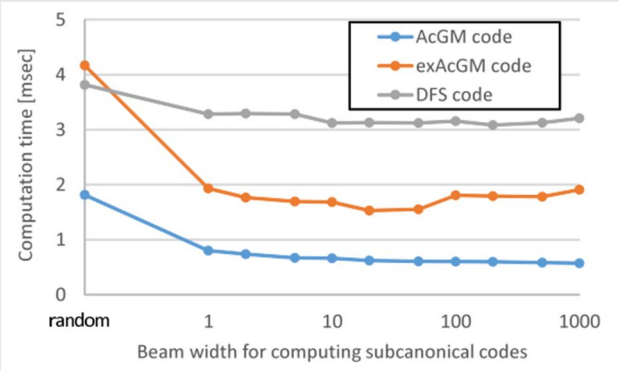

図 2 正準形探索のビーム幅と検索時間

図 2 は包摂グラフ検索問題を提案した手法 CodeTree で解いたときの実験結果である．データベースに含まれるグラフは，約 4 万個の化合物であり，クエリは前述の化合物に含まれていた辺数が 25 の化合物である．CodeTree では，グラフをグラフコードで表すが，1 つのグラフにつき，複数のグラフコードが対応する．それらのコードのうち最大のコードを正準形と呼ぶが，正準形（canonical code）を求める問題は大きな計算時間を要するので，CodeTree ではビーム幅をある値に設定して準正準形(sub-canonical code)を求めている．横軸で示されるビーム幅に設定したときの検索時間が縦軸に示されている．ビーム幅を 100 程度にすると，1 クエリにつき約数ミリ秒でグラフ検索できた結果が左図に示されている．

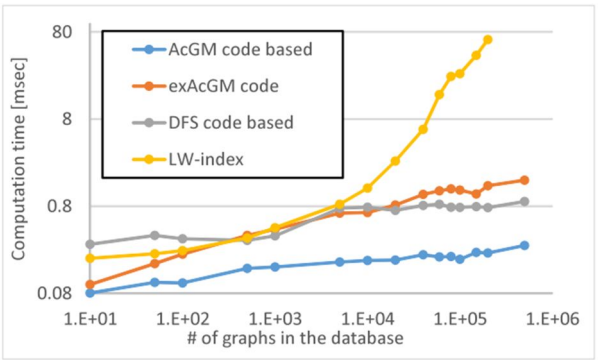

図 3 データベース内のグラフ数と検索時間 1

図 3 は包摂グラフ検索問題を解いたときの結果である．横軸はデータベース内のグラフ数であり，上記の約 4 万グラフの部分グラフである．LW-Index [7]は開発当初に最速であったアルゴリズムである．それ以外の折れ線は，提案手法 CodeTree の結果であり，グラフ数の増加とともに検索時間は増加するか，LW-Index に比べ CodeTree は大規模なデータベースに適用できることが見てとれる．近年，DGTree[2]や IDAR[1]など LW-Index よりも高速な手法が提案されている．これらの性能を上回る手法を開発することが今後の課題である．

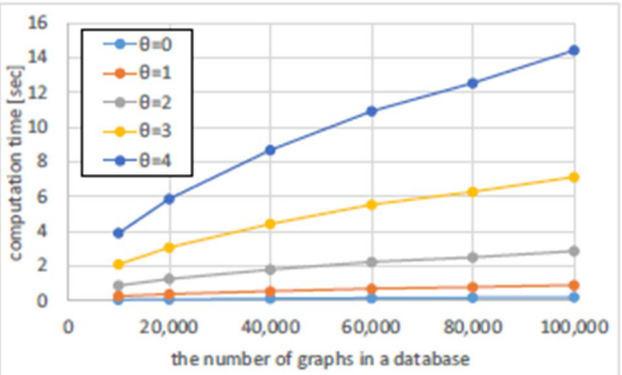

図 4 データベース内のグラフ数と検索時間 2

図 4 は，類似包摂グラフ検索問題を CodeTree で解いたときの実験結果である．θは，クエリグラフの部分グラフとグラフ内のグラフ gi との編集距離の閾値である．グラフは化合部やその部分グラフであり，データベース内のグラフの平均頂点数は 29.0，クエリグラフの平均頂点数は 71.0 である．データベース内のグラフの数を増やすと，検索時間は増加する．また，θを増やすと計算時間は増加する．θを増やすことで，編集数θで書き換えることができるグラフの数は，指数関数的に増加するが，それによって解として出力されるグラフ数やそ

の候補もまた指数関数的に増加するからである．このため，θを 5 以上にすると計算爆発が起こるが，実際の運用を考えると q が任意に編集されたグラフを検索するよりも，q の特定の頂点や辺が編集されたグラフの検索のほうが使いやすい．そのような検索ユーザの要望が容易に指定できることも，文献[6]で示している．

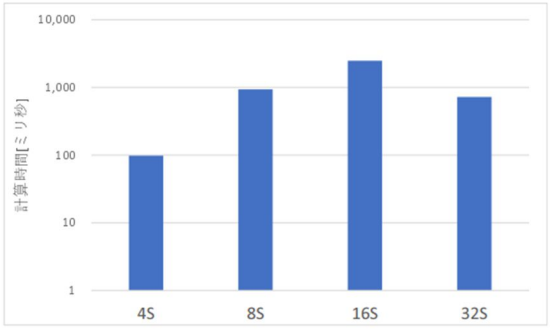

図 5 部分グラフ検索の計算時間

図 5 は，部分グラフ検索に CodeTree を適用した場合の結果である．実験設定は，文献[4]と同じ，4 万個の化合物がデータベースに格納されている．また，そこから 4 つの辺からなる部分グラフがクエリグラフとして選ばれ，それを用いて実験した結果が 4S として示されている．8S，16S，32S も同様である．文献[4]では CFQL が提案されているが，左の結果は CFQL と同等，クエリ条件によっては CFQL よりも高速な実験結果が得られた．今後，我々の CodeTree の動作を詳細に調査し，その結果を公表していく予定である．

参考文献

[1] Hyunjoon Kim, Seunghwan Min, Kunsoo Park, Xuemin Lin, Seok-Hee Hong, and Wook-Shin Han: IDAR: Fast Supergraph Search Using DAG Integration. Proc. VLDB Endowment. 13(9), pp. 1456-1468, 2020.

[2] Bingqing Lyu, Lu Qin, Xuemin Lin, Lijun Chang, and Jeffrey Xu Yu. Scalable Supergraph Search in Large Graph Databases. Proc. of IEEE International Conference on Data Engineering, pp. 157-168, 2016.

[3] 白木公康．緊急寄稿（2）新型コロナウイルス感染症（COVID-19）治療候補薬アビガンの特徴．週刊日本医事新報，5005 号，pp. 25，2020.

[4] Shixuan Sun and Qiong Luo: Scaling Up Subgraph Query Processing with Efficient Subgraph Matching. Proc. of. IEEE International Conference on Data Engineering, pp. 220-231, 2019.

[5] 山田真賢，猪口明博.グラフ編集距離を用いた類似包摂グラフ検索．情報処理学会データベースシステム．2020.

[6] Masataka Yamada and Akihiro Inokuchi: Similar Supergraph Search Based on Graph Edit Distance. Algorithms 14(8): 225, 2021.

[7] Dayu Yuan, Prasenjit Mitra, and C. Lee Giles. Mining and Indexing Graphs for Supergraph Search, Proc. VLDB Endowment, 6(10), pp.829-840, 2013.

５．主な発表論文等

〔雑誌論文〕　計2件（うち査読付論文　2件／うち国際共著　0件／うちオープンアクセス　1件）

１．著者名 Masataka Yamada, Akihiro Inokuchi	４．巻 14
２．論文標題 Similar Supergraph Search Based on Graph Edit Distance	５．発行年 2021年
３．雑誌名 Algorithms	６．最初と最後の頁 -
掲載論文のDOI（デジタルオブジェクト識別子） 10.3390/a14080225	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

１．著者名 Shun IMAI, Akihiro INOKUCHI	４．巻 E103.D
２．論文標題 Efficient Supergraph Search Using Graph Coding	５．発行年 2020年
３．雑誌名 IEICE Transactions on Information and Systems	６．最初と最後の頁 130-141
掲載論文のDOI（デジタルオブジェクト識別子） 10.1587/transinf.2019EDP7011	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

〔学会発表〕　計4件（うち招待講演　0件／うち国際学会　1件）

１．発表者名 大西 悠平, 猪口 明博
２．発表標題 カットオフ値が未知の多変量データに対する回帰不連続デザイン
３．学会等名 人工知能学会 知識ベースシステム研究会
４．発表年 2021年

１．発表者名 Yuta Yajima, Akihiro Inokuchi
２．発表標題 Refining Similarity Matrices to Cluster Attributed Networks Accurately
３．学会等名 5th International Workshop on GPU Computing and AI（国際学会）
４．発表年 2020年

１．発表者名 山田真賢，猪口明博
２．発表標題 グラフ編集距離を用いた類似包摂グラフ検索
３．学会等名 情報処理学会 データベースシステム研究会
４．発表年 2020年

１．発表者名 辻川拓摩，猪口明博
２．発表標題 確率的なリラベルを用いたグラフ分類の精度向上
３．学会等名 情報処理学会 データベースシステム研究会
４．発表年 2020年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関