グループ単位の関連度評価に基づく知識グラフ探索アルゴリズムの設計と評価

八尾　正剛† 　福井　健太† 　LiGuangcan† 　鬼塚　真†

† 大阪大学　〒 565–0871 大阪府吹田市山田丘 1–1

E-mail: †{yao.seigo,fukui.kenta,guangcan.li,onizuka}@ist.osaka-u.ac.jp

あらまし　LLM の推論精度を向上する技術として，GraphRAG における知識グラフ上での情報検索の重要性が高まっている．既存研究では，与えられた質問を構成する各質問エンティティをクエリとして検索して得られるノード集合をグループとして表現し，各グループから少なくとも 1 つのノードを含み，かつノード間の接続コストの総和が最小となるサブグラフを決定する問題として Group Steiner Tree（GST） に基づく手法が広く用いられている．しかし，GST の目的関数は接続コストの最小化に限定されているため，質問エンティティに対するノードの適合度という本質的な情報を活用したサブグラフ選択を行うことができない．そこで本研究では，各グループに属するノードの適合度を単調劣モジュラ関数で集約した値を目的関数に組み込むことで，本問題を定式化する．さらに，総当たりでは実行時間が長くなるため，事前計算と複数の評価指標に基づく段階的な局所探索を導入し，高速に近似解を探索する．実験では，3 つのデータセットと複数の設定における実験において，既存手法よりも高い精度を達成した．

キーワード　知識グラフ，局所探索法

1　序　　論

近年，知識グラフを活用し，グラフ上の関係をもとに情報を検索・統合して検索結果や質問応答の形で関連情報を提示する知識集約型タスクが注目されている． [1,2]．このような知識グラフ活用の枠組みは，知識グラフを用いた情報検索 [3,4] や推薦 [5] に加え，医療・バイオ分野の質問応答 [6]，サイバーセキュリティにおける情報分析・質問応答 [7]，法務領域の文書 QA [8]，製造業の故障モード影響解析 [9] など様々な分野で応用されている．

しかし，知識集約型タスクで用いられる知識グラフは一般に大規模であり，入力クエリに関連する情報はグラフ全体に散在するため，クエリに適合する根拠サブグラフをどのように選択するかが，検索精度と計算効率の両面から重要な課題となっている [3,10,11]．

実運用のクエリには，対象となるエンティティを指す語句に加えて，エンティティ間に求める関係や条件を表す語句が併存し，複数のエンティティ・関係・制約を同時に含むことが多い [12]．そのため，各クエリ要素に対応する候補ノードのクエリ要素への適合度を考慮しつつ，要素ごとの取りこぼしが生じないように各要素を被覆する必要がある [11,13]．このようなトレードオフを定量的に扱うため，根拠サブグラフ抽出はグラフ上の最適化問題として定式化される [14–16]．

根拠サブグラフ抽出の代表的な定式化として，質問を構成する質問エンティティごとに対応する候補ノード集合をグループとして定義し，すべてのグループを被覆しつつ接続コストが最小となる部分グラフを求める Group Steiner Tree（GST）が知られている [13,16]．GST はクエリを構成する要素を必ず含むという制約を表現できる一方で，目的関数が接続コストの最小化に限定されるため，候補ノードの質問エンティティへの適合度を目的関数に直接取り込めないという課題がある．また，ノードに質問との適合度を表す重みを prize として与え，接続コストとのトレードオフを最適化することが可能な定式化として Prize-Collecting Steiner Tree（PCST） [17,18] が知られている．しかし，PCST ではグループ被覆を制約に含んでいないため，クエリにおける要素が欠落してしまう場合がある．更に，ノードの重みと接続コストの両方を最適化する新たな定式化の方法として Node-Weighted Group Steiner Tree (NW-GST) が知られている [19]．しかし，NW-GST ではノード重みを線形加算でしか評価することができないため，同一グループ内でノードを追加で選ぶことによる利得は選択済みのノードに依存せず常に一定となり，情報検索分野で知られている限界逓減性質 (既に選択した情報と類似した情報を追加しても得られる価値が小さくなる性質) [20] を表現できないという問題がある．

そこで本研究では，そのような限界逓減性質を単調劣モジュラ関数として目的関数に組み込むことで，本問題を定式化する．具体的には，各グループに対して解に含まれるノードの適合度を，単調劣モジュラ関数で集約し，これを接続コストと同時に最適化することで，グループ被覆を満たしつつ，各グループ内で高適合度ノードを優先して選択する根拠サブグラフ抽出を実現する．また，本問題を単純な総当たりで厳密に解くことは計算量の観点から現実的ではないため，本研究では目的関数を近似した複数の評価指標に基づく段階的な局所探索を導入し，探索空間を絞り込みながら高速に近似解を求める手法を提案する．

本稿の貢献を以下にまとめる．

- 知識集約型タスクにおける根拠サブグラフ抽出に対し，グループ被覆とグループ内適合度を単調劣モジュラ関数で集約した値を，同時に扱う最適化問題を定式化した．
- 総当たり探索の計算量的困難さに対し，複数の評価指標を

用いた段階的な局所探索により，高速に近似解を探索するアルゴリズムを提案した．
- 複数データセット・複数設定において実験を行い，精度と実行時間の両立という観点から提案手法の有効性を検証した．

2 関連研究

本研究が扱う根拠サブグラフ抽出は，関連度の高いノードの選択とそれらの構造的接続コストを同時に最適化する Steiner Tree 系の組合せ最適化問題の拡張として位置付けられる．このタイプの問題は，知識グラフ探索，キーワード検索，QA などで広く用いられている．代表的な定式化として，グループ被覆制約を持つ Group Steiner Tree（GST），ノード適合度とエッジコストのトレードオフを扱う Prize-Collecting Steiner Tree（PCST），およびノード重みを考慮する Node-Weighted Group Steiner Tree（NW-GST）が知られている．以下では，これらの定式化を整理し，特性と限界を述べる．

Group Steiner Tree（GST）： GST は，グラフ上でグループとして与えられる複数の候補集合をそれぞれ少なくとも 1 点ずつ含み，エッジ重みの和が最小となる木を求める問題である．クエリ検索では，クエリ語ごとに類似したノード集合をグループとみなし，それらを結ぶ最小コストの木を返す定式化が古典的に用いられてきた [10, 14, 21]．

知識グラフ探索の分野でも，クエリに関連するエンティティを結ぶ根拠サブグラフを，GST や topk-GST から求める手法が用いられている [10, 22]．また，QA では，複数の文書から抽出した関係をノード・エッジとして構成した擬似的な知識グラフから，GST を用いてクエリに対応した複数の根拠を接続し，回答生成するシステムが提案されている [13, 23]．

Prize-Collecting Steiner Tree（PCST）： PCST は，各ノードにクエリに対する適合度に応じた prize，各エッジにコストが与えられたときに，含まれるノードの適合度の和とエッジコストのトレードオフが最適化されるような木を求める問題である．Graph RAG では，大規模グラフから LLM に投入可能なサイズまで情報を圧縮しつつ，適合度の高いノードをなるべく多く含み，かつ構造的にまとまったサブグラフを抽出する目的で PCST 定式化を採用したものが知られている [11]．また，PCST は，RAG 用途のサブグラフ抽出でも利用されている．例えば，対話システム向けに，文脈関連度を報酬として与え，最小サイズで最大関連のサブグラフを得るために PCST として解く手法が提案されている [24]．また，マルチモーダル知識グラフ QA でも，関連性と構造的一貫性を担保するサブグラフ抽出として PCST が用いられている [25]．

Node-Weighted Group Steiner Tree（NW-GST）： NW-GST は，GST のようなグループ被覆制約を保ちつつ，エッジ重みに加えてノード重みもコストとして取り込み，ノードとエッジのコストの総和が最小となる木を求める問題である．このような設定は，ノードそのものにコストが設定されるような課題で有用である [26]．

NW-GST のアルゴリズムの研究としては，GST における既存近似手法を拡張したアルゴリズムなどを与える研究が知られている [19]．NW-GST は，リレーショナル DB からの情報検索や，ソーシャルネットワーク，知識グラフからの情報抽出に用いることができる [19]．

既存定式化の課題: 既存の定式化である GST，PCST，NW-GST には，それぞれ次のような課題がある．GST はノードの適合度を目的関数に含んでいないため，ノードの適合度も考慮した上でクエリに対して最適な木を選択することができない．PCST はグループ被覆を制約として含んでいないため，各クエリ要素に対する情報の取りこぼしが発生することがある．また，NW-GST はグループごとに選択したノードの適合度を単純な加算で評価するため，情報検索において基本的な考えである限界逓減性質を考慮した木を選択することができない．これらの課題を解決するため， 4 章で新たな定式化を提案する．

3 事前知識

知識グラフを $G = (V, E)$ と表す．ここで，知識グラフは無向グラフであり，V はノード集合，E はエッジ集合である．各エッジ $e \in E$ には非負の接続コスト $c(e) \in \mathbb{R}_{\geqq 0}$ を割り当て，サブグラフ $T = (V_T, E_T)$ の接続コストを

$$\mathrm{Cost}(T) = \sum_{e \in E_H} c(e) \tag{1}$$

と定義する．また，ノード $u, v \in V$ 間の最短距離は $d(u, v)$ と表し，ノード v から集合 H への距離を $d(v, H) = \min_{u \in H} d(v, u)$ と書く．$d(v, H)$ は multi-source dijkstra を用いて求める [27]．

a） 質問エンティティに対応するノードグループ

質問 $\boldsymbol{q}$ は，m 個の質問エンティティから構成されるとする．各質問エンティティ q_i $(1 \leqq i \leqq m)$ に対して，類似度関数 $\mathrm{sim}(\cdot, \cdot)$ に基づき，対応する候補ノード集合をノードグループ G_i として

$$G_i = \mathrm{TopK}(V,\ v \mapsto \mathrm{sim}(q_i, v),\ k)$$

と定義する．このとき，質問エンティティ q_i とノード v の間で算出される適合度 $\mathrm{sim}(q_i, v)$ を，$q[i]$ に関するノード v の prize と呼び，一般的にサブクエリとノードの埋め込み類似度や BM25 のスコア等から導出される．

b） グループ被覆条件

質問を構成する全ての質問エンティティを網羅した根拠サブグラフ T を選択するために，以下で定義されるグループ被覆を条件として導入する．

$$V_T \cap G_i \neq \emptyset \qquad (\forall i \in \{1, \ldots, m\}) \tag{2}$$

この条件は各グループから少なくとも 1 ノードを含むことに相当し，質問を構成する全ての質問エンティティに対応する根拠の欠落を抑える目的で導入される．

このように，入力となる知識グラフ $G = (V, E)$，グループ G_i，グループ被覆条件を定義する．また，グループ G_i に対するノード $v \in V$ の prize を $p_i(v)$ とする．

4 提案手法

本研究では，グループ被覆制約を満たしつつ，各グループ内で適合度の高いノードを優先して選択し，かつノード間の接続コストを同時に考慮する根拠サブグラフ抽出問題を定式化する．但し，これらを同時に最適化する組合せ最適化問題は計算量的に困難であり，厳密解法は大規模グラフでは現実的でない．そこで本研究では，実用的な計算時間内で高品質な解を得ることを目的として，計算効率と解品質の両立を図る．具体的には，制約付き組合せ最適化で広く用いられる局所探索を用いてグループ被覆制約を保ったまま解を逐次改善するアプローチを採用し，局所探索における初期解の構成および評価機構を以下の方針に基づいて具体化する．(1) prize が高く，かつ他のグループに近い有望なノードを各グループごとに選択し，初期解とする．(2) 局所探索では，ノードの適合度と現在の選択集合との距離の両方を考慮した有望度指標に基づいて探索候補を優先順位付けする．(3) 解の評価においては，厳密な Steiner 木コストの計算は高コストであるため，軽量な近似評価からより厳密な評価へと移行する 2 段階の評価指標を用いる．

以降，4.1 節で根拠サブグラフ抽出問題を定式化し，4.2 節で，高速に近似解を得るための局所探索アルゴリズムを述べる．

4.1 提案問題の定式化

本節では，入力として与えられる知識グラフ，質問エンティティに対応するノードグループ，およびノード適合度に基づき，本研究で扱う根拠サブグラフ抽出問題を定式化する．

本研究の目的は，(i) 式 (2) で定義されたグループ被覆制約を満たしつつ，(ii) 各グループに対して高い適合度をもつノードを優先的に含み，(iii) 接続コストが大きくならない根拠サブグラフ $T = (V_T, E_T)$ を抽出することである．

まず，prize に関する条件 (ii) と接続コストに関する条件 (iii) を表現する目的関数を以下のように定義する．

定義 1 (目的関数)**.** 連結サブグラフ $T = (V_T, E_T)$ に対し，目的関数 $F(T)$ を以下のように定義する．

$$F(T) = \mathrm{Cost}(T) - \lambda \sum_{i=1}^{m} S_i(T) \tag{3}$$

但し，$\lambda > 0$ は各グループ G_i に対する寄与度 $S_i(T)$ の総和と接続コストのトレードオフを制御する係数である．各グループ i と候補ノード $v \in G_i$ に対して，クエリ q_i との適合度（prize）を

$$p_i(v) := \mathrm{sim}(q_i, v) \tag{4}$$

と定義する．さらに，本研究では単調劣モジュラ関数 f を導入し，$S_i(T)$ を以下で定義する：

$$S_i(T) = f(\{\, \Sigma_{v \in V_T \cap G_i} p_i(v) \}). \tag{5}$$

式 (5) は，同一グループ内で複数ノードを追加したときの寄与が限界逓減するよう，単調劣モジュラ性に基づいて評価を集約する点に特徴がある．具体的には，任意の集合 $A \subseteq B$ と要素 $x \notin B$ に対して

$$S_j(A \cup \{x\}) - S_j(A) \;\geqq\; S_j(B \cup \{x\}) - S_j(B) \tag{6}$$

が成り立つため，既に多くのノードが選ばれているほど追加ノードの増分は小さくなる．この性質により，不要なノードが T に混入することを抑制しつつ，各グループの情報を段階的に補うことが可能である．

次に，グループ被覆条件 (i) を満たすことを制約として，上記の目的関数の最適化問題を以下のように定式化する．

定義 2 (サブグラフ抽出の最適化)**.** $G = (V, E)$ を与えられたグラフとする．本研究では，G の連結サブグラフ $T = (V_T, E_T)$ に対し，次の最適化問題を考える．

$$T^\star = \arg\min_{T \subseteq G} F(T) \tag{7}$$

$$\text{s.t.} \quad V_T \cap G_i \neq \emptyset, \qquad \forall i \in \{1, \ldots, m\}, \tag{8}$$

$$T \text{ is connected.}$$

4.2 提案アルゴリズム

本節では，定義 2 で示したサブグラフ抽出の最適化問題に対し，高速に近似解を得るための局所探索アルゴリズムを述べる．本アルゴリズムは，目的関数に対してそのノードがどれだけ有望かを数値化して探索候補を決定する指標に基づく初期選択，二段階評価に基づく局所探索，および一括再選択による再初期化，得られたターミナル集合によるサブグラフの構築から構成される．以降では，各グループ G_i から選択されるターミナルノードの集合を $V_T = \{r_1, \ldots, r_m\}$（$r_i \in G_i$）と表わす．

4.2.1 初期解の選択

本アルゴリズムは局所探索を行うため，初期解の選択が最終的に得られる解の品質に大きく影響する．このため定義 1 で示した目的関数に基づいて，各グループごとに prize が高く，かつ他グループとの距離が近い有望なノードを初期解として選択することが重要である．有望なノードを選択する指標として，グループ G_j におけるノード v のスコア

$$B_j(v) = S_j(v)/P_{\max} + \beta A_j(v) \tag{9}$$

を導入し，各グループ G_j ごとに $r_j = \arg\max_{v \in G_j} B_j(v)$ となるノード r_j を初期解として選択することで，初期解 $V_T \leftarrow \bigcup_{j=1}^{m} \{r_j\}$ を得る．但し，右辺の第一項で使用されている

$$P_{\max} = \max_j \max_{v \in G_j} f_j(\{v\})$$

であり，prize を正規化するために用いる．また，第二項の $A_j(v)$ は他グループへの近さを近似する指標であり，以下のように定義される．

$$A_j(v) = \frac{1}{m-1} \sum_{\ell \neq j} \exp\!\left(-\frac{D_\ell(v)}{\sigma}\right) \tag{10}$$

σ には $\{D_\ell(v) \mid v \in \bigcup_j G_j,\ \ell = 1..m\}$ の中央値を用いる．指数関数を用いることで，極端に大きな $D_\ell(v)$ の影響を小さくしつつ，ノード v の他のグループ l への近さを近似する．

4.2.2 二段階評価に基づく局所探索

ここでは，現在のターミナル集合 V_T に対し，以下の 3 種類の近傍操作を反復する．

- **ADD**(b)：$V_T \leftarrow V_T \cup \{b\}$,
- **REMOVE**(a)：$V_T \leftarrow V_T \setminus \{a\}$,
- **EXCHANGE**(a,b)：ADD(b) の後に REMOVE(a).

a） 探索候補の絞り込み

各反復ごとに，目的関数の値を改善する可能性の高い有望なノードを探索候補集合 B とする．このような目的関数に対して有用度が高いノード v は，現在のターミナル集合 V_T に加えた場合の木のスコアの増分が大きく，かつ V_T への距離が小さい方が望ましいことから，PIA スコアを次のように設計する．

$$J_j(v,V_T) = \alpha \frac{\Delta S_j(v,V_T)}{P_{\max}} + \beta \exp(-d(v,V_T)/\sigma_T), \quad (11)$$

$$\Delta S_j(v,V_T) = S_j\big(V_T \cup \{v\}\big) - S_j\big(V_T\big).$$

但し，$S_j(V_T) = V_T \setminus (V_T \cap G_j)$ であり， σ_T には $d(\cdot, V_T)$ の中央値を用いる．式 (11) の右辺の第 1 項はノード v を V_T に加えた場合の木のスコアの増分近似であり，第 2 項はノード v の V_T に対する近さを表したものであるため，式 (11) の値はノード v の V_T に対する有望度を表す．そして，PIA スコアが各グループで上位 t 以内のノードの集合を探索候補集合 B とする．

b） 代理評価を用いたフィルタリング

式 (1) で示されるサブグラフの接続コストを厳密に計算するには，シュタイナー木の計算が必要となるため，探索候補集合 B が与えられたときに，考えられる近傍操作全てに対して式 (1) 計算すると高コストである．そこで本アルゴリズムでは，安価な代理評価によって改善の見込みが高い操作のみを残す．

ノード $v \in G_j$ を V_T に加える操作 ADD(b) については，木コストの増分を

$$\widehat{\Delta C}_{\text{add}}(b) = d(b, V_T)$$

で近似し，寄与度増分は b を含む全グループの限界利得を足し合わせて

$$\widehat{\Delta P}_{\text{add}}(b) = S_j(V_T \cup \{b\}) - S_j(V_T)$$

とする．そして

$$\widehat{\Delta F}_{\text{add}}(b) = \widehat{\Delta C}_{\text{add}}(b) - \widehat{\Delta P}_{\text{add}}(b) \quad (12)$$

の小さい候補 b を少数残す．

REMOVE(a) については，現在のターミナル集合 V_T と $V_T \setminus \{a\}$ の目的関数値を次の $\widehat{F}(V_T)$ で粗く見積もる．

$$\widehat{F}(V_T) = \widehat{C}(V_T) - \sum_{j=1}^{m} S_j(V_T), \widehat{C}(V_T) = \min_{\ell} \sum_{t \in V_T} D_\ell(t) \quad (13)$$

そして $\widehat{F}$ の差分 $\widehat{F}(V_T') - \widehat{F}(V_T)$ を現在のターミナル集合 V_T からノード a を取り除いたときのスコアの減少幅とみなし，その値が大きい候補 a を少数残す．EXCHANGE についても同様に，現在のターミナル集合 V_T と $(V_T \setminus \{a\}) \cup \{b\}$ に対する式 (13) の差分を計算し，その値が大きい候補 (b,a) を少数残す．EXCHANGE は ADD と REMOVE の組み合わせとして表現できるが，ADD，REMOVE を個別に行った場合では目的関数値が改善しない場合でも，EXCHANGE として同時に実行すれば目的関数が改善することがあり得るため導入する．

c） 厳密評価と受理

代理評価で残った少数の候補に対してのみ，式 (1) の値を実際に計算し，F を最も改善する操作を 1 つだけ受理して V_T を更新する．同様の反復を最大反復回数 R まで，あるいは改善が得られなくなるまで繰り返す．

4.2.3 再 初 期 化

4.2.2 節の局所探索は，目的関数の値が改善する場合のみ解を更新し，解の改悪を許容しない．そのため，探索範囲が狭まり局所解に陥る可能性が高くなっている．この問題を緩和するため，局所探索後に初期解の再選択を行い，その初期解を元に局所探索をもう一度行うことで探索範囲を大幅に変更する．現在のターミナル集合を V_T として，各グループ j に対し 1 ノードを再選択するスコアを

$$I_j(v) = \alpha \frac{S_j(\{v\})}{P_{\max}} + \beta A_j(v) + \eta A_j^{\text{term}}(v; V_T) \quad (14)$$

と定義する．$A_j(v)$ は (10) と同様に他グループへの近さを表す項であり，$A_j^{\text{term}}(v;V_T)$ は現在のターミナル群への近さを表す項であり，

$$A_j^{\text{term}}(v;V_T) = \frac{1}{|S_j(V_T)|} \sum_{t \in S_j(V_T)} \exp(-d(v,t)/\sigma_T) \quad (15)$$

である．これにより，各グループで $I_j(v)$ 最大の候補を選び直して新しい V_T を作り， 4.2.2 節の局所探索を再実行し，最適なターミナル集合 V_T^* を探索する．

4.2.4 出　　力

以上のようにして，得られたターミナル集合 V_T^* をターミナルとして，既存の Steiner tree 近似アルゴリズム [28] を実行することで最終的なサブグラフを出力する．

5 実　　験

提案手法の優位性を示すため，以下に示す 3 つの問いを設定して評価実験を実施した．

実験 1: 提案手法は，既存の根拠サブグラフ抽出と比べて，グループ被覆を満たしつつ高い適合度の根拠を選択できるか？

実験 2: 提案アルゴリズムの構成要素はそれぞれ，精度にどの程度寄与するか？

実験 3: ハイパーパラメータは，解の傾向にどのような影響を与えるか？

5.1 実 験 設 定

データセット: 本研究では，Cora，CiteSeer，PubMed の 3 つの引用ネットワークデータセットを用いる [29]．いずれも論文をノード，引用関係を辺として表現した単一グラフからなり，各ノードには Bag-of-Words に基づく特徴量とトピックラベルが付与されている．これらのノード数，エッジ数，特徴量数は

データセット	ノード数	エッジ数	特徴量数
Cora	2,708	5,429	1,433
CiteSeer	3,327	4,732	3,703
PubMed	19,717	44,338	500

表 1: データセット統計

表 1 のようになっている [30].

グループ構築: m をグループ数，k は各グループが含むノード数とする．グループを構築するために，グラフ上でランダムなノードを m 個選択し，それらをシードノードする．これらシードノードとの類似度が上位 k 以内であるノードをそれぞれのシードノードを元にしたグループとする．

prize 設定: 各グループ G_j に対してグループ内のノードの，シードノードとの類似度に基づく順位を $r_j(v) \in \{1, \ldots, |G_j|\}$ で表す．このとき，グループ G_j におけるノード v の基礎 prize を

$$\tilde{p}_j(v) = \begin{cases} |G_j| - r_j(v) & (v \in G_j), \\ 0 & (v \notin G_j) \end{cases} \tag{16}$$

と定義する．実験では，この基礎 prize に式 (1) 中でのスケール係数 $\lambda > 0$ を乗じた

$$p(v) = \lambda \tilde{p}(v) \tag{17}$$

を最終的な prize として用いる．ここで，λ は目的関数における prize による利得と接続コストの相対的な重要度を制御するパラメータである．λ を大きくすると，高 prize ノードを得ることを優先して多少の遠回りを許容する傾向が強まり，逆に λ を小さくすると，グラフ構造を優先して prize の差を相対的に無視する傾向が強まる．したがって，λ の設定は抽出される根拠サブグラフの性質に大きく影響する．

比較手法: 提案手法の有効性を検証するため，以下の代表的な手法と比較する．

- Max-Prize：各グループ G_i から prize が最大のノードを 1 つずつ選択し，それらをターミナルとして Mehlhorn の Steiner 木近似により接続する [28].
- PCST（pcst_fast）：PCST の近似実装である `pcst_fast` を用いる [31]．ただし，PCST は標準形ではグループ被覆制約を持たないため，各グループを被覆できるようにダミーノードを追加する変換を施し，変換後の PCST を解くことでグループ制約を実現する．
- NW-GST（exENSteiner）：Node-weighted GST の既存アルゴリズムである `exENSteiner` を用いる [19]．グループごとに劣モジュラ関数で集約された適合度を，サブグラフのスコアとして加算するという提案問題における目的関数を再現するため，ダミーノードを追加する変換をグラフに施した上で解く．具体的には，まず各グループ G_i で単体での prize が最大のノードを 1 つ選び，その後はすでに選んだ集合 S_i に追加したときの増分

$$f(S_i \cup \{v\}) - f(S_i)$$

が最大となる候補を順に最大 3 個まで選ぶことで，あるノードを選択したときの利得 $\Delta_i(v)$ を段階的に算出する．残りの候補は，この 3 個の集合に対する増分

$$\Delta_i(v) = f(S_i \cup \{v\}) - f(S_i)$$

により近似する．そして，各候補 v に対してダミー葉 v^* を追加して v と 0 コストの辺で接続する．v の重みを 0 とし，ダミー葉の重みを

$$w(v^*) = M - \Delta_i(v)$$

と設定することで，$\Delta_i(v)$ が大きい候補ほど重みが小さくなり，解として選ばれやすくなる．

- NW-GST（FastAPP）：NW-GST（exENSteiner）と同様に，ダミーノードを用いたグラフの変換をした上で，NW-GST の `FastAPP` を用いる [19].
- NW-GST（ImprovAPP）：NW-GST（exENSteiner）と同様に，ダミーノードを用いたグラフの変換をした上で，NW-GST の `ImprovAPP` を用いる [19].

評価指標: 評価指標として目的関数の値および平均順位，勝率を用いる．提案問題は最小化問題であるため，目的関数値が小さいほど良い解である．また，あるアルゴリズムの順位としては，その実験設定において最も目的関数の値が良い解を探索したアルゴリズムの順位を 1 とし，それ以外のアルゴリズムの順位を降順でつける．なお，目的関数の値が同じであった場合は同じ順位をつける．勝率は，複数回の実験において順位が 1 であった割合とする．

ハイパーパラメータ: 実験では，グループサイズ k を 30 ，グループ数 m を $\{4, 8, 12, 16\}$ ，式 (1) 中でのスケール係数である λ を $\{0.5, 1.0, 2.0, 4.0\}$ に変化させ，各実験設定ごとにシードを 8 通りに変えて実験を行った．単調劣モジュラ関数としては，代表的な *log* 関数，*max* 関数，*sqrt* 関数，*topk* 関数を用いて実験した [32, 33]．提案手法における局所探索の最大反復回数 R は 20 とした [1].

実験環境: 全ての実験は，Ubuntu 20.04.6 LTS 上のサーバ（HPE ProLiant DL385 Gen10 Plus）で行った．提案アルゴリズムの実装については Python で行い，NW-GST の実装は C++ で行った，主要ライブラリとして NumPy，SciPy，NetworkX，scikit-learn，pcst_fast を用いた．

5.2 実　験　1

実験 1 では，提案手法が，既存の根拠サブグラフ抽出と比べて，グループ被覆を満たしつつ高い適合度の根拠を選択できるかを確認するため，パラメータを $\lambda = 1.0$，$m = 8$，$f = \log$ に固定して実験し，既存手法と提案手法の比較を行った．実験結果を表 2 に示す．それぞれの値は 8 回の実験の結果を平均したものである．$\lambda = 1.0$ は，本研究と同様に prize とコスト両方の最適化をすることで根拠サブグラフを抽出する既存研究を

1：早期終了した実験の 95%が 17 ラウンド以内で終了したため $R = 20$ と設定した．

	Cora		Citeseer		Pubmed	
手法	目的関数値 ± 標準偏差	平均順位	目的関数値 ± 標準偏差	平均順位	目的関数値 ± 標準偏差	平均順位
Max-Prize	0.67 ± 4.48	6.00	9.11 ± 4.07	6.00	4.62 ± 4.13	5.25
pcst_fast	-6.09 ± 1.77	4.12	-5.54 ± 1.44	3.25	-3.43 ± 1.98	2.38
exENSteiner	-9.49 ± 2.92	2.69	-5.33 ± 3.36	3.81	-0.61 ± 2.65	3.81
FastAPP	-5.46 ± 3.05	4.25	-2.36 ± 4.70	4.00	5.74 ± 5.36	5.25
ImprovAPP	-9.45 ± 2.30	2.81	-6.60 ± 3.14	2.69	-1.27 ± 2.57	3.06
提案手法	**-13.62 ± 2.22**	**1.12**	**-8.91 ± 1.89**	**1.25**	**-6.37 ± 1.98**	**1.25**

表 2: 固定したパラメータにおける手法ごとの目的関数値と標準偏差，および平均順位

手法	Cora	Citeseer	Pubmed
Max-Prize	4.02	4.09	4.04
pcst_fast	5.53	5.31	5.22
exENSteiner	3.01	3.11	2.72
FastAPP	4.17	4.13	4.79
ImprovAPP	2.71	2.71	2.64
提案手法	**1.56**	**1.64**	**1.59**

表 3: データセット別の平均順位．最も高い精度のものを太字にしている．

踏まえて設定した [11]．$m = 8$ は，本研究での比較手法として用いている NW-GST の既存研究での実験設定を踏まえて設定した [19]． $f = \log$ は，本研究と同様に，劣モジュラ関数を用いて効用を表現する既存研究を踏まえて設定した [32]．また，データセットの解の品質に対する影響を確認するため，データセットごとでの既存手法と提案手法の比較を行った．その結果を表 3 に示す．それぞれの値は 512 回の実験の結果を平均したものである． 表 2 から，提案手法が比較手法に対して高い精度，および安定性を両立していることがわかる．これは，提案手法が既存手法と異なり，グループごとの prize を集約する劣モジュラ関数を考慮して解を探索できているからだと考えられる．また，表 3 からすべてのデータセットにおいて提案手法が比較手法よりも高い精度を達成しており，データセットに関わらず，精度を維持できることがわかる．これは，提案手法が PIA スコアにおいて prize と辺コストを正規化しているため，データセットとして用いたグラフのサイズに影響されないからだと考えられる．

5.3 実 験 2

実験 2 では提案アルゴリズムの構成要素がそれぞれ，精度にどの程度寄与するかを確認するため，提案手法の構成要素を取り除いた場合の実験を行った．具体的には，初期解の選択において単純に各グループごとに prize が最大のノードを選んだもの，PIA スコアを使わず prize 順で探索候補集合 B を選んだもの，再初期化を実行しなかったものの計 3 種類を提案手法と比較する．実験結果を表 4 に示す．それぞれの値は 128 回の実験の結果を平均したものである．

手法	平均順位	勝率
提案手法 w/o 初期化	2.16	52.9%
提案手法 w/o PIA	1.67	69.9%
提案手法 w/o 再初期化	1.66	72.1%
提案手法	**1.59**	**74.8%**

表 4: Pubmed における ablation 実験の結果．最も精度の高いものを太字にしている．

表 4 から提案手法の構成要素それぞれが解の精度向上に貢献していることがわかる．特に，初期解選択を工夫することで精度が大きく向上しており，局所探索における初期解選択の重要性が確認できる．

5.4 実 験 3

実験 3 では，ハイパーパラメータが解の傾向にどのような影響を与えるかを確認するため，スケール係数 λ，グループ数 m，劣モジュラ関数 f それぞれを変更して実験を行った．実験結果を表 5，表 6，表 7 に示す．それぞれの値は 128 回の実験の結果を平均したものである．

手法	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 2.0$	$\lambda = 4.0$
Max-Prize	4.46	4.09	3.92	3.68
pcst_fast	4.51	5.13	5.52	5.73
exENSteiner	2.86	2.85	2.61	2.57
FastAPP	4.94	4.86	4.74	4.63
ImprovAPP	2.80	2.63	2.57	2.57
提案手法	**1.43**	**1.46**	**1.65**	**1.83**

表 5: Pubmed におけるスケール係数 λ 別の平均順位．最も精度の高いものを太字にしている．

手法	$m=4$	$m=8$	$m=12$	$m=16$
Max-Prize	4.11	3.99	4.14	3.91
pcst_fast	5.02	5.08	5.40	5.38
exENSteiner	2.63	2.93	2.62	2.71
FastAPP	4.70	4.90	4.70	4.86
ImprovAPP	2.86	2.63	2.48	2.59
提案手法	**1.68**	**1.48**	**1.67**	**1.55**

表 6: Pubmed におけるグループ数 m 別の平均順位．最も精度の高いものを太字にしている．

手法	*log*	*max*	*sqrt*	*topk*
Max-Prize	4.96	3.45	4.17	3.57
pcst_fast	3.43	6.00	5.45	6.00
exENSteiner	3.22	1.94	2.67	3.06
FastAPP	5.30	4.83	4.98	4.05
ImprovAPP	2.78	**1.85**	2.62	3.31
提案手法	**1.31**	2.93	**1.13**	**1.00**

表 7: Pubmed における劣モジュラ関数 f 別の平均順位．最も精度の高いものを太字にしている．

表 5，表 6 はそれぞれスケール係数，グループ数を変更した場合の既存手法と提案手法の平均順位を示す．それら表から，提案手法がスケール係数，グループ数に関わらず一貫して高い精度を発揮していることがわかる．表 7 は劣モジュラ関数を変更した場合の実験結果を示す．*topk* 関数では $k=3$ とした．表 7 では f として max 関数を用いた場合に既存手法に比べて精度が低いことが確認できる．これは，max 関数が限界逓減効果の極端に大きい関数であり，一つのグループに対して複数のノードを解のサブグラフに加えることが目的関数値を悪化させるが，提案アルゴリズムではそのような極端な限界逓減効果に対応することが難しいからだと考えられる．同じく限界逓減効果が比較的強い関数である *log* 関数を使った場合にも，*sqrt* や *topk* に比べて精度が低くなっている．

6 結　　論

本研究では，グループ被覆を満たしつつ，各グループ内で高適合度のノードを優先して選択する根拠サブグラフ抽出問題を定式化，およびその問題に対する近似解法を提案し，その精度を検証した．提案問題は，劣モジュラ関数を導入することで，グループごとの情報の網羅度を定量的に扱い，限界逓減効果を目的関数として表現した．また，提案アルゴリズムは，有望な初期解の選択，PIA スコアによる探索空間の絞り込み，再初期化によって高い精度での根拠サブグラフ探索を実現した．

今後の展望としては，再初期化前の局所探索の簡素化や出力におけるサブグラフ構築の高度化に取り組み，より高速に高精度な根拠サブグラフ抽出ができるよう，提案アルゴリズムの改善および拡張に取り組む予定である．

謝　　辞

本研究は JSPS 科研費 JP25H01117 の助成を受けたものです。

文　　献

[1] Dennis Diefenbach, Vanessa Lopez, Kamal Singh, and Pierre Maret. Core techniques of question answering systems over knowledge bases: A survey. *Knowledge and Information Systems*, 55(3), 2018.

[2] Arnaldo Pereira, Alina Trifan, Rui Pedro Lopes, and José Luís Oliveira. Systematic review of question answering over knowledge bases. *IET Software*, 16(1):1–13, 2022.

[3] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. Knowledge graphs: An information retrieval perspective. *Foundations and Trends® in Information Retrieval*, 14(4):289–444, 2020.

[4] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *ACL*, 2018.

[5] Qijun Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems. *TKDE*, 34(8):3549–3568, 2022.

[6] Ying Huang et al. A knowledge graph based question answering method for medical domain. *PeerJ Computer Science*, 7:e667, 2021.

[7] Ruby Rani, Mahender Kumar, Gregory Epiphaniou, and Carsten Maple. ICSThreatQA: A knowledge-graph enhanced question answering model for industrial control system threat intelligence. *Expert Systems with Applications*, 301:130180, 2026.

[8] Francesco Sovrano, Monica Palmirani, and Fabio Vitali. Legal knowledge extraction for knowledge graph based question-answering. In *JURIX*, pages 143–153, 2020.

[9] Lukas Bahr, Christoph Wehner, Judith Wewerka, José Bittencourt, Ute Schmid, and Rüdiger Daub. Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis. *Journal of Industrial Information Integration*, 45:100807, 2025.

[10] Fan Yang et al. Keyword search on large graphs: A survey. *Data Science and Engineering*, 2021.

[11] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *NeurIPS*, 2024.

[12] Yunshi Lan, Jing Jiang, Xin Jiang, Wayne Xin Zhao Wang, and Ji-Rong Wen. A survey on complex knowledge base question answering: Methods, challenges and solutions. In *IJCAI*, pages 4483–4491, 2021.

[13] Xiaolu Lu, Soumajit Pramanik, Rishiraj Saha Roy, Abdalghani Abujabal, Yafang Wang, and Gerhard Weikum. Answering complex questions by joining multi-document evidence with quasi knowledge graphs. In *SIGIR*, pages 105–114, 2019.

[14] Wenjie Li, Yinan Qin, Jeffrey Xu Yu, and Rong Mao. Efficient and progressive group steiner tree search. In *SIGMOD*, 2016.

[15] Aaron Archer, MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Howard Karloff. Improved approximation algorithms for prize-collecting Steiner tree and TSP. *SIAM Journal on Computing*, 40(2):309–332, 2011.

[16] Naveen Garg, Goran Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group steiner tree

problem. *Journal of Algorithms*, 37(1):66–84, 2000.

[17] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[18] David S. Johnson, Maria Minkoff, and Steven Phillips. The prize-collecting steiner tree problem: Theory and practice. In *SODA*, pages 760–769, 2000.

[19] Yahui Sun, Xiaokui Xiao, Bin Cui, Saman K. Halgamuge, Theodoros Lappas, and Jun Luo. Finding group steiner trees in graphs with both vertex and edge weights. *Proc. VLDB Endow.*, 14(7):1137–1149, 2021.

[20] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[21] Bolin Ding, Jeffrey Xu Yu, Shan Wang, Lu Qin, Xiao Zhang, and Xuemin Lin. Finding top-k min-cost connected trees in databases. In *ICDE*, pages 836–845, 2007.

[22] Yunliang Shi, Jinhua Zhuang, Yu Zhang, Qifan Wang, Changyou Chen, Yaliang Li, Jun Yuan, and Jiawei Han. Keyword-based knowledge graph exploration based on quadratic group steiner trees. In *IJCAI*, 2021.

[23] Soumajit Pramanik, Jesujoba Alabi, Rishiraj Saha Roy, and Gerhard Weikum. Uniqorn: Unified question answering over RDF knowledge graphs and natural language text. *Journal of Web Semantics*, 83:100833, 2024.

[24] Nicholas Thomas Walker, Pierre Lison, Laetitia Hilgendorf, Nicolas Wagner, and Stefan Ultes. Retrieving relevant knowledge subgraphs for task-oriented dialogue. In *SIGDIAL*, pages 513–526, 2025.

[25] Z. Ali, A. Haldar, K. Korhonen, and J. Kontio. Pythia-rag: Retrieval-augmented generation with unified multimodal knowledge graph. *Knowledge-Based Systems*, 335:115200, 2026.

[26] Mehdi Kargar, Aijun An, and Morteza Zihayat. Efficient bi-objective team formation in social networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 483–498, 2012.

[27] Xiang Wei, Wei Lu, and Weiwei Xing. A rapid multi-source shortest path algorithm for interactive image segmentation. *Multimedia Tools and Applications*, pages 21547–21563, 2017.

[28] Kurt Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27(3):125–128, 1988.

[29] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, pages 40–48, 2016.

[30] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[31] Chinmay Hegde, Piotr Indyk, and Ludwig Schmidt. A nearly-linear time framework for graph-structured sparsity. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pages 928–937, 2015.

[32] Aida Rahmattalabi, Shahin Jabbari, Himabindu Lakkaraju, Phebe Vayanos, Max Izenberg, Ryan Brown, Eric Rice, and Milind Tambe. Fair influence maximization: A welfare optimization approach. In *AAAI*, pages 11630–11638, 2021.

[33] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *ACL*, pages 510–520, 2011.