4L-04

大規模グラフにおける到達可能性クエリ問題に対する頂点へのレベル付け方法

岩瀬 宇延† 久保田 光一‡

中央大学大学院 理工学研究科 情報工学専攻†‡

1 はじめに

ソーシャルネットワークや Web グラフ, 生体ネットワークなどにおいて, 大規模グラフの頂点間の関連性を解析する場合などでグラフの到達可能性クエリは大量に要求されるため, 各クエリに対して高速に回答することが求められる.

そこで, そのような大規模グラフにおける到達可能性クエリに対して高速に回答する手法として, 予め頂点へレベル付けをしラベルの構築を行い, その構築したラベルを用いて到達可能性クエリに回答するという Zhu 等の手法がある [2]. この手法では, 頂点へのレベル付けに大きく依存し, レベル付けの方法によって大きく性能が異なってくる.

本研究では, Zhu 等の方法 [2] に対していくつかの方法により頂点へのレベル付けを行い, 各レベル付けによる性能を比較することにより, どのグラフに対してどのようなレベル付けがより良いかの検証を行う.

2 用語

2.1 レベル

グラフ G 上の全ての頂点に対して, $1 \sim |V|$ までの順位を付けていく. このとき, 各頂点 v に付けられた順位をレベルといい, $l(v)$ と表す. また, レベルは $l(v)$ が小さい方をレベルが高い, 大きい方をレベルが低いという.

2.2 ラベル

ラベルは各頂点 v に $L_{in}(v)$ と $L_{out}(v)$ の 2 種類が対応し, $L_{in}(v)$ は頂点 v の祖先の部分集合, $L_{out}(v)$ は頂点 v の子孫の部分集合となっている.

各頂点 v のラベルの要素 $^{\forall}u \in L_{in}(v)$(resp. $L_{out}(v)$) は以下の 3 つの制約を満たす.

到達制約:
$u \to v$(resp. $v \to u$).

レベル制約:
$l(u) < l(v)$.

パス制約:
頂点 u から v (resp. 頂点 v から u) への全てのパス上において, $l(w) < l(u)$ を満たす頂点 w は存在しない.

ラベルサイズは, 全頂点のラベルの要素数の総和で表し, $|L| = \sum_{v \in V}(|L_{in}(v)| + |L_{out}(v)|)$ と定義される.

3 アルゴリズム [2]

3.1 ラベル構築

有向無閉路グラフ G に対するラベルの構築方法を Algorithm 1 に示す.

Algorithm 1 ラベル構築

```
1: G_1 ← G
2: for k ← 1 to |V| do
3:    レベルが k の頂点を v_k とする
4:    頂点 v_k から G_k 上において幅優先探索を行い, 訪れた
      頂点の集合を B^+(v_k) とする
5:    for u ∈ B^+(v_k) do
6:       if L_out(v) ∩ L_in(u) = ∅ then
7:          L_in(u) ← L_in(u) ∪ {v}
8:       end if
9:    end for
10:   グラフ G_k の全ての辺の向きを逆向きにしたグラフを
      G'_k とし, 頂点 v_k から G'_k 上において幅優先探索を行
      い, 訪れた頂点の集合を B^-(v_k) とする
11:   for u ∈ B^-(v_k) do
12:      if L_out(u) ∩ L_in(v) = ∅ then
13:         L_out(u) ← L_out(u) ∪ {v}
14:      end if
15:   end for
16:   グラフ G_k から頂点 v_k を取り除き, 取り除いたグラフ
      を G_{k+1} とする
17: end for
```

3.2 クエリ回答

構築したラベルを用いて到達可能性クエリに回答する方法を説明する. 頂点 s から t への到達可能性クエリに対して, 式 (1) のように定義される QUERY(s,t) を計算しクエリに回答する. つまり, $(L_{out}(s) \cup \{s\}) \cap (L_{in}(t) \cup \{t\})$ が空でなければ True, すなわち頂点 s から t へ到達可能であり, そうでなければ False, すなわち到達不可能であると回答する.

$$\mathrm{QUERY}(s,t) = \begin{cases} \text{True}, & (L_{out}(s) \cup \{s\}) \cap (L_{in}(t) \cup \{t\}) \neq \emptyset; \\ \text{False}, & \text{otherwise.} \end{cases} \tag{1}$$

4 レベル付け方法

4.1 InOut

頂点 v の入次数 を $d_{in}(v)$, 出次数を $d_{out}(v)$ とする. 各頂点 v を $(d_{in}(v)+1) \times (d_{out}(v)+1)$ の値の降順にレベル付けを行う方法を InOut とする.

4.2 Upper bound & Lower bound [2]

Upper bound, Lower bound 共に, 初めにグラフ G 上において, 式 (2) の $f(v,G)$ の値が 1 番大きい頂点 v_1 のレベルを $l(v_1) = 1$ とする. その後, グラフ G から頂点 v_1 を削除し, そのグラフを G' とし, 再度 $f(v,G')$ を計算する. 次にグラフ G' 上において, $f(v,G')$ の値が 1 番大きい頂点 v_2 のレベルを $l(v_2) = 2$ とする. この作業を全ての頂点を削除するまで繰り返し行う.

$$f(v,G) = \frac{S_{in}(v,G) \times S_{out}(v,G)}{S_{in}(v,G) + S_{out}(v,G)}. \tag{2}$$

$S_{in}(v,G)$ と $S_{out}(v,G)$ は Upper bound, Lower bound そ

Leveling Vertex for Reachability Queries on Large Graphs

† Takanobu IWASE, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

‡ Koichi KUBOTA, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

れぞれにおいて以下のように定義される．なお，$N_{in}(v)$ と $N_{out}(v)$ はそれぞれ頂点 v の入近傍と出近傍を表している．

Upper bound:

$$S_{in}(v,G) = \sum_{u \in N_{in}(v)} S_{in}(u,G) + 1,$$
$$S_{out}(v,G) = \sum_{u \in N_{out}(v)} S_{out}(u,G) + 1.$$

Lower bound:

$$S_{in}(v,G) = \sum_{u \in N_{in}(v)} \frac{S_{in}(v,G)}{|N_{out}(v,G)|} + 1,$$
$$S_{out}(v,G) = \sum_{u \in N_{out}(v)} \frac{S_{out}(v,G)}{|N_{in}(v,G)|} + 1.$$

4.3 Static Upper bound & Static Lower bound

Upper bound と Lower bound では 1 つの頂点にレベルを付ける度にグラフからその頂点を削除し，$f(v,G)$ の値を更新していた．Static Upper bound, Static Lower bound は，Upper bound, Lower bound それぞれにおいて値の更新を行わず $f(v,G)$ の値の大きい順にレベル付けを行う方法である．

5 実験

本実験では全てのアルゴリズムの実装を C++11 で行い，gcc 4.8.2 を用いて最適化オプション -O3 の設定でコンパイルを行い，CPU が Intel Xeon E5-1650 v3@3.50GHz，メモリが 128GB の Linux サーバー上にて実験を行った．本実験において使用したデータセットは表 1 の通りである．なお，表 1 の頂点数および辺数は SNAP[1] のグラフデータに対して，各強連結成分において縮約を行い，有向無閉路グラフにしたものとなっている．平均応答時間は 1,000,000 回の到達可能クエリの応答時間に対する平均を取っている．

表 1　グラフデータ

データセット	$\|V\|$	$\|E\|$
email-EuAll	231,000	223,004
web-Google	371,764	517,805
soc-LiveJournal1	971,232	1,024,140
wiki-Talk	2,281,879	2,311,570
cit-Patents	3,774,768	16,518,947

5.1 実験結果

ラベル構築時間 (図 1) を見ると，全てのグラフデータにおいて Upper bound, Static Upper bound によるレベル付け方法が，他のレベル付けの方法と比べ良い性能を示している．特に，cit-Patents においては，かなり優れた性能を示している．次に，ラベルサイズ (図 2)，平均応答時間 (図 3) を見てみると，ラベル構築時間と同様に Upper bound, Static Upper bound によるレベル付けが良い性能を示していることがわかる．最後に，レベル構築時間 (図 4) を見ると，全てのグラフデータにおいて InOut, Static Upper bound, Static Lower bound がほぼ同程度で良い性能を示している．

これらのことから，ラベル構築時間・ラベルサイズ・平均応答時間では全てのグラフデータにおいて Upper bound, Static Upper bound によるレベル付けが他の方法と比べより優れており，レベル構築時間では InOut, Static Upper bound, Static Lower bound が優れていることから，Static Upper bound によるレベル付けが一番良い性能を示すことがわかる．

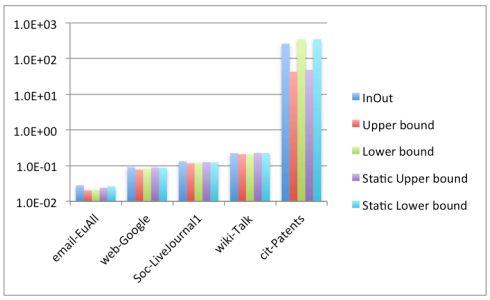

図 1　ラベル構築時間 (sec)

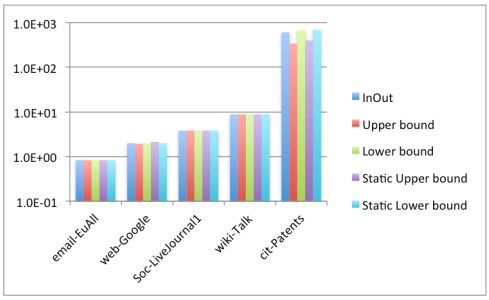

図 2　ラベルサイズ (MB)

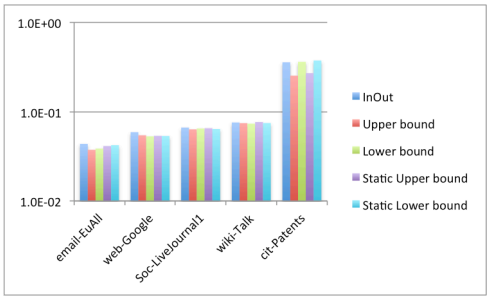

図 3　平均応答時間 (μ sec)

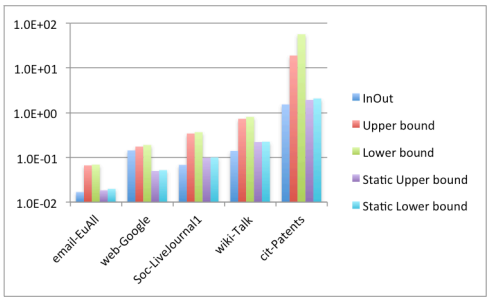

図 4　レベル構築時間 (sec)

参考文献

[1] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[2] Andy Diwen Zhu, Wenqing Lin, Sibo Wang, and Xiaokui Xiao. Reachability queries on large dynamic graphs: A total order approach. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pp. 1323–1334, 2014.