グラフ・サンプリングにおけるグラフ探索アルゴリズムの選択

王　キン† 成　凱‡

†‡九州産業大学大学院 情報科学研究科 〒813-8503 福岡市東区松香台 2-3-1

E-mail: †k12gjk01@st.kyusan-u.ac.jp, ‡chengk@is.kyusan-u.ac.jp

あらまし 近年、複雑な構造をもつデータが急増し、グラフ構造を扱うためのデータ解析が重要になっている。グラフデータを解析するためにランダムサンプリング手法がよく用いられ、グラフから代表的部分を抽出して巨大分散グラフでも素早く分析できる。これまでの研究ではサンプリングの重要な一環となるグラフ探索、およびサンプル更新は深く研究されず、ランダムウォークを基本とするサンプリング手法を用いるのはほとんどである。例えば、ランダムウォークを用いたサンプリング手法として、PageRank に基づくランダムウォーク、ForestFire、 MRW (Metropolized Random Walk)等が知られている。本研究では、我々はグラフサンプリング手法を分析し新しいサンプリング手法を設計するため、グラフ探索、サンプル抽出、サンプル更新を含む枠組みを提案する。特に、これまで深く研究されなかったグラフ探索アルゴリズムの選択、サンプル更新の仕組みを考察し、実験による評価を行う。

キーワード グラフ探索アルゴリズム、グラフサンプリング、ランダムウォーク

On the selection of graph exploring algorithms for graph sampling

Xin Wang† and Kai Cheng‡

†‡Graduate School of Information Science, Kyushu Sangyo University

3-1, Matukadai 2-chome, Higashi-ku, Fukuoka, 813-8503

E-mail: †k12gjk01@st.kyusan-u.ac.jp, ‡chengk@is.kyusan-u.ac.jp

1. はじめに

近年、社会学、生命科学、ビジネス分析、ソーシャルネットワーク等の分野で、複雑な構造をもつデータが急増し、グラフデータとしてその重要性が広く認識されている。例えば、ソーシャルネットワークサービスにおける利用者同士の交友関係、ビジネス分野における企業間の取引関係、オンラインショッピングサイトにおける顧客商品関係、また、社会学の分野においては、噂の広がり、意見形成、疫病の拡散など、グラフを扱うためのデータ解析が必要不可欠になっている。大規模データを解析するため、一部代表的なデータを無作為に抽出するランダムサンプリング手法がよく用いられる。無作為に抽出されるサンプルでは、母集団のデータがすべて同じ出現確率で出現することが望まれる。

しかし、大規模なグラフデータ対するサンプリングでは、無作為性を保証することが難しいと知られている。まず、グラフデータが巨大で集中管理されていない場合が多い。例えば、WWW や P2P ネットワーク。そのため、グラフ構造を探索しながら、既知の頂点から隣接頂点へアクセスしながらサンプルを集めければならない。グラフ構造やグラフ探索方法によって、集められたサンプルは偏りや相関性等の問題が生じる可能性が高い。例えば、次数の高い頂点ほど抽出される確率が高い。隣接する頂点はともに抽出される確率が高い。

これまでは様々なサンプリング手法やアルゴリズムが提案された[9][13][15][16]。これらの提案では基本的にランダムウォーク（Random Walk）を用いてグラフを探索し、経由した頂点をサンプルとして抽出している。ランダムウォークとは、グラフ上である頂点からランダムに隣接頂点を一つ選んでそこに移動し、移動先からまた同じ方法で次の頂点を選び移動を続けるグラフ探索の仕組みである。ランダムウォークは現在頂点の情報しか必要とせず、探索のコストが低い利点がある一方、次数の高い頂点に偏ったり、連結度の高い部分グラフにとどまったり、グラフ全体をカバーする時間が長くカバー率が低い問題点が指摘されている[15][16]。

本研究では、我々はグラフサンプリングの枠組みとして**グラフ探索、サンプル抽出、サンプル更新**の三つ

の構成要素を提案する。特に、これまで深く研究されなかったグラフ探索アルゴリズムの選択、サンプル更新の仕組みについて考察し、実験による評価を行う。

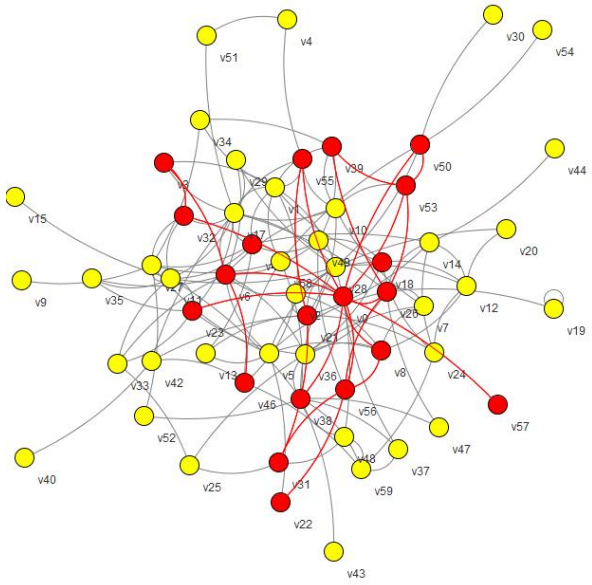

図 1　グラフ・サンプリング例（赤）

2.　グラフサンプリングにおけるグラフ探索

グラフの頂点を対象とするサンプリング手法は、**図 2** に示すように、グラフ探索、サンプル抽出、サンプル更新を含む次のような枠組みで考察できる。

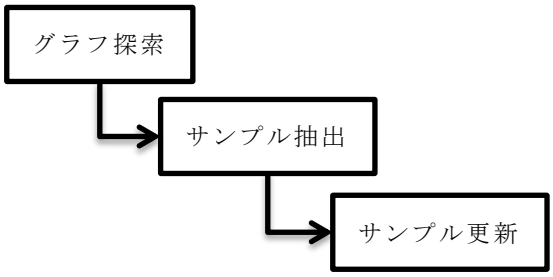

図 2　グラフサンプリングの流れ

1. **グラフ探索**：ある頂点から次の移動先の頂点を選び、グラフ構造を探索しながら、データを収集する手法。
2. **サンプル抽出**：グラフ探索で得られた頂点をサンプルとして抽出する方法。探索で集めた頂点はすべて抽出するか、それとも一定の基準に従い一部を抽出するか、その判断を行う。
3. **サンプル更新**：探索範囲の拡大に伴い、新たに抽出されたサンプルを用いて既存のサンプルを更新する方法。グラフ構造がすべて事前に把握することができない場合に、探索が進むと同時に、サンプルも適切に更新する必要がある。

グラフ探索は、移動先の選び方によって分類でき、移動先を決定的に選ぶ**決定的グラフ探索**と、移動先を確率的に選ぶ**確率的グラフ探索**と呼ぶ。また、同じ頂点が複数回訪問される可能があるかどうかによって、**重複なしグラフ探索**、**重複可能なグラフ探索**がある。

決定的グラフ探索方法として、幅優先探索（BFS: Breadth First Search）、深さ優先探索（DFS: Depth First Search）がよく知られている。未展開頂点の保存やバックトラックのために別途メモリが必要である。メモリ使用量を抑えるため、深さ制限探索（DLS: Depth Limited Search）、反復深化深さ優先探索（IDDFS: Iterative Deepening Depth First Search）等が提案されている。決定的探索では訪問済み頂点を二度と訪問しないので、重複なしグラフ探索である。

幅優先探索(BFS)とはグラス上で与えられた頂点(木構造の場合は木の根）で始まり隣接した全ての頂点を訪問する。それからこれらの最も近い頂点のそれぞれに対して同様のことを繰り返して訪問範囲を拡大していく。未訪問頂点を訪問するまで FIFO キューに格納しておく必要がある。

深さ優先探索　(DFS)は、グラフ上で指定の頂点で始まり、バックトラックするまで可能な限り探索を行い、それぞれの頂点を訪問する。その後はバックトラックして、最も近くの未訪問の頂点まで戻ってまた同様な方法を繰り返す。未訪問頂点を訪問するまで LIFO キュー（スタック）に格納する必要がある。

深さ制限探索(DLS)は深さ優先探索からの派生で、探索する深さを制限し、制限を越えて探索を深くしていくことがないため、無限の経路に捉われたり、環状の経路に捉われたりすることがない。DFS はグラフをカバーするまで深く探索しないため、次に述べる反復深化深さ優先探索の一部として利用される。

反復深化深さ優先探索（IDDFS）とは、深さ制限探索の制限を徐々に増大させ、最終的に目標状態の深さになるまで反復するものである。各反復では深さ優先探索の順序で探索木の頂点を調べるが、全体として見れば（刈り込みがない場合)、各頂点を初めて調べる順序は幅優先探索と同じ順序になる。

確率的グラフ探索方法として、ランダムウォークを基本として様々な改良方法がある。ランダムジャンプ（Random Jump)、ForestFire (BFS で展開する頂点数を乱数で決める方法)[8]等がある。確率的グラフ探索では、同じ頂点を複数回訪問する可能性があり、重複可能なグラフ探索でもある。

また、ランダムウォークの改良として、MRW (Metropolized Random Walk)がよく知られている[8]。MRW では、サンプリング結果の均一な分布を得るために、Metropolis-Hastings 法を用いて遷移確率を修正している。

$$\varrho(x,y) = \begin{cases} p(x,y)\,min\left(\frac{deg(x)}{deg(y)},1\right), x \neq y \\ 1 - \sum_{x \neq y} \varrho(x,y), \qquad x = y \end{cases}$$

具体的には、無向グラフにおいて x が隣接ノードから無作為（等確率）に任意の y を候補遷移先として選ぶ。しかし本当に y に遷移するかはさらに確率 Q(x, y)で決める。候補遷移先 y の次数が x の次数より大きいなら、遷移確率 Q(x, y)を小さめに調整する。候補遷移先 y の次数が x の次数より小さいなら、遷移確率 Q(x, y)を調整しない。このように、次数の高いノードに集中する傾向を抑えることができる。

グラフ探索はサンプリングの効率に影響が大きい。探索方法によって、グラフをカバーする時間(Cover Time)が違う。カバー時間とはグラフ上のある頂点から始め、すべて頂点を訪問するまでにかかる移動数（step）の期待値である。例えば、頂点数 n の連結グラフをすべてカバーするために、単純ランダムウォークのカバー時間の上界が $O(n^3)$であると知られている[1]。グラフの一部を訪問する場合は、一定の移動数でカバーするグラフ範囲が大きいほどよい。また、探索経路を記憶するに必要なメモリ容量はできれば少ないほうがよい。

表 1 は頂点数 n、辺数 m、幅 b、深さ d の無向グラフを探索する各種方法の比較である。z は Forest Fire 法で展開する最大頂点数、実際に展開するのは z 以下の乱数である。

表 1 グラフ探索手法の比較

探索手法	種類	カバー時間	空間計算量
RandomWalk	確率的	$O(n^3)$	O(1)
ForestFire	確率的	-	$O(z^d)$
BFS	決定的	O(n+m)	O(bd)
DFS	決定的	O(n+m)	O(bd)
IDDS	決定的	$O(b^d)$	O(bd)

3. サンプルの抽出と更新

サンプル抽出では、グラフ探索で得られた頂点をサンプルとして抽出するかどうかを決める。代表的抽出方法には、無作為抽出（Random Sampling)、系統抽出（Systematic Sampling)、層化抽出（Stratified Sampling）がある。

1. **無作為抽出**：探索された頂点の一部（または全部）をランダム（無作為）に抽出する。
2. **系統抽出**：探索された頂点に通し番号をつけておき、はじめのサンプルだけランダムに選び、あとは一定間隔（インターバル）で系統的に抽出する。
3. **層化抽出**：探索された頂点をいくつかの層（群）に分け、そこから一定の比率でサンプルを抽出する。

上記の手法で抽出したサンプルを維持し、探索の進行に伴い、必要に応じて適切に更新する。これまでの研究では、サンプル更新についてあまり議論されず、既存のサンプルを削除せずにすべて残すことが一般的である。しかし、動的グラフや規模が予知できないグラフに対して、グラフ探索が進むとともにサンプルが更新される必要がある。以下のサンプル更新手法が考えられる。

1. **スライディングウィンドウ（Sliding Window）を用いたサンプル更新**。新しいサンプルが入ることにより古いサンプルを削除し、サンプルを入れ替える。
2. **Reservoir を用いたサンプル更新**。新しいサンプルと古いサンプルの抽出確率を同様に保つため、Reservoir を用いる。

Reservoir を用いたサンプリング（Reservoir Sampling）は母集団のデータが一度しかアクセスできない場合や、新しいデータが無限に出てくる場合に、データ出現の順序に関係なく等確率でサンプルを抽出する手法である[7]。サンプルサイズを k として事前にきめておく。母集団から k 番目までのデータはそのままサンプルに入る。そして、i 番目（i>k）以降のデータが到来したとき、[1..i]の範囲で乱数 r を生成し、r は k より小さい場合のみ、新しいデータをサンプルの r 番目として保存し、元のサンプルが上書きされて消える。R が k より大きい場合は新しいデータを無視し Reservoir はそのまま変わらない。Reservoir を利用する際に、サンプルサイズ（Reservoir 容量）を事前にきめる必要がある。

4. グラフサンプリングアルゴリズム

これまでの議論からわかるように、適切なグラフサンプリング手法を考案するために、グラフ探索、サンプル抽出、サンプル更新を含む各要素を考慮しなければならない。

本章ではサンプル更新に用いる手法によって 2 種類のグラフサンプリングを提案する。アルゴリズムを説明するために、**表 2** に示すような記号を使う。関数

Move()と Sample()は探索方法、サンプル抽出方法に依存し、詳細は具体的にアルゴリズムで決める。

表 2 アルゴリズム説明に使う記号

記号	説明
G<V, E>	探索対象となるグラフ
S	探索開始点の集合
k	サンプルサイズ
MaxStep	探索実行の最大ステップ数
ArrayList	サンプル一時保存用配列
Reservoir	サンプル一時保存用 Reservoir
CURR	現在訪問中の頂点
Move()	探索順に従い次へ移動（CURR を更新）
Sample()	CURR を抽出するかを決める

4.1. Reservoir を用いたグラフサンプリング手法

Reservoir を用いたグラフサンプリング（GSR : Graph Sampler with a Reservoir）を紹介する。**図 3** に示すように、グラフ探索で次から次へ移動しながら、Sample()で決められた方針でサンプルを抽出する。抽出された頂点を Reservoir に格納しサンプルを更新する。

アルゴリズム GSR

入力：

(1) G<V, E>、S、k、MaxStep

(2) Reservoir[1…k]: サンプル一時保存用 Reservoir

出力:

抽出結果：部分グラフ G'=(V', E') （サンプル）

処理手順：

1. Reservoir 初期化；step = 1;
2. 始点集合 S から任意一つをランダムに選び CURR として探索開始
3. IF (CURR==NULL)：
   ステップ２へ移動し探索再開
4. decision = Sample();
5. IF (decision==拒否)：
   ステップ７へ移動
6. IF (step<=k) Reservoir[step]=CURR;
   ELSE
   [1..step]から乱数 r を振る
   IF (r <= k) Reservoir[r]=CURR;
7. Move(); step = step + 1;
8. IF (step ≥ MaxStep)：
   G'=(V', E')を出力して終了
   ELSE：
   ステップ３へ戻る

図 3 Reservoir を用いたグラフサンプリング

GSR は静的グラフをサンプリングする場合に適している。動的グラフでは特に頂点やグラフ構造が激しく変化する場合に適用できない。

4.2. Sliding Window を用いたグラフサンプリング

進化し続けるグラフに対し、スライディングウィンドウを用いたグラフサンプリング（GSW： Graph Sampler with a Sliding Window）を提案する。アルゴリズムの概要は**図 4** に示している。

アルゴリズム GSW

入力：

1. G<V, E>、S、k、MaxStep
2. ArrayList[1…k]：サンプル一時保存用配列

出力:

抽出結果：部分グラフ G'=(V', E') （サンプル）

処理手順：

3. ArrayList 初期化；step = 1; rear =1
4. 始点集合 S から任意一つをランダムに選び CURR として探索開始
5. IF (CURR==NULL) ：
   ステップ２へ移動し探索再開
6. decision = Sample();
7. IF (decision==拒否)：
   ステップ７へ移動
8. IF (step<=k)：
   ArrayList[rear]=CURR; rear =rear+1;
   IF (rear>k) rear = 1;
9. Move(); step = step + 1;
10. IF (step ≥ MaxStep)：
    G'=(V', E')を出力して終了
    ELSE：
    ステップ３へ戻る

図 4 Sliding Window を用いたグラフサンプリング

グラフ探索で得られた頂点は Sample()によって抽出の判断を行う。拒否すれば、サンプルとして残せず探索が続く。抽出する場合は、rear 番の配列要素として格納する。rear は末尾を超えると、また先頭に戻り、そこに最も古いテータがあり、新しいサンプルを格納することにより、古いサンプルが消される。

5. 実験評価

これまで述べてきたグラフ探索とサンプル更新手法の組み合わせを評価するために評価実験を行う。実験では、代表的なグラフをランダムグラフとして生成し、抽出結果の次数分布を中心に評価を行う。

実験用のシミュレーターは、JUNG (Java Universal Network/Graph Framework)を用いて実装した。JUNG は Java でグラフ構造の生成、処理、可視化等ができるオープンソースのライブラリーである[18]。

5.1. 評価尺度

グラフサンプリングにおけるグラフ探索アルゴリズムを評価するために、以下の尺度を用いる。

1. 偏差（Error）：抽出されたサンプルの次数分布が基準となる次数分布と比べて偏差の度合い。
2. 効率（Efficiency）：サンプルを集めるまで探索などに必要な時間やメモリ容量。

母集団の要素がサンプルとして平等に選ばれることで望ましい。一部の要素が他より選ばれやすい場合に偏りが生じる。偏ったサンプルは一般に誤った推定に与える。ここで以下の三つの尺度を用いてサンプルの次数分布と基準となる次数分布を比較し、その偏差を評価する。

a. カルバック・ライブラー情報量（Kullback–Leibler divergence：**KL**）：2 つの確率分布の差異を計る尺度。離散確率分布 P の Q に対するカルバック・ライブラー情報量は以下のように定義される。

$$D_{\mathrm{KL}}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

b. 対称カルバック・ライブラー情報量（Symmetric Kullback–Leibler divergence：**SK**）

c. 平均二乗誤差（Squared Error：**SE**）：2 つの確率分布 P と Q の差異を以下の式で評価する。

$$D_{SE}(P,Q) = \frac{1}{n}\sum_i (P(i) - Q(i))^2$$

5.2. 評価対象

表 3 では評価対象となる具体的アルゴリズムを示している。サンプル更新手法として、Reservoir を RV、Sliding Window を SW と略称する。

表 3 評価対象のアルゴリズム

	サンプル更新手法		従来手法
探索手法	RV	SW	BN
BFS	BFS-RV	BFS-SW	BFS-BN
DFS	DFS-RV	DFS-SW	DFS-BN
RW	RW-RV	RW-SW	RW-BN
MRW	MRW-RV	MRW-SW	MRW-BN

サンプル抽出の従来手法として Bernoulli サンプリング（BN と略称）も実装した。一定の確率で探索し集めてきた頂点をサンプルとして抽出する。

また、実験に使うグラフは以下の三種類のランダムグラフを生成し、それぞれに対して、各サンプリングアルゴリズムを使って、サンプルを抽出し、評価を行う。

Uniform：Erdös-Rényi ランダムグラフ。任意の頂点対が一定確率で隣接するグラフである。

PowerLaw：Eppstein ランダムグラフ。次数分布がべき乗則に従う。

SmallWorld：Kleinberg ランダムグラフ。スモールワールド（Small World）の性質をもつグラフ。つまり、辺の数が頂点数の数倍程度しかないが、頂点間の平均距離が小さくて高度にクラスター化している。

抽出対象グラフ G<V, E>について以下の共通のパラメターをもつ。

- 頂点数：n=|V| = 22,500
- 辺数：m=|E|=151,8750, (n×n×0.003)
- 探索開始点集合 S：s=|S|=22, (n×0.001)
- サンプル抽出率 p：
  p = { 0.05, 0.10, 0.15, 0.2, 0.25, 0.30, 0.35};
- 最大探索移動数 q=225,000, (n×10)。無限ループに陥らないようにこの数以上探索しない。

偏差を評価するためには基準となる次数の確率分布を用いる。本研究では、元のグラフにおける次数分布を基準とする。また、理想なサンプルとして、グラフ構造を考慮せず、すべての頂点を平等に抽出する。このような理想的状況は現実的には存在しないので Oracle を呼ぶ。

5.3. 評価結果

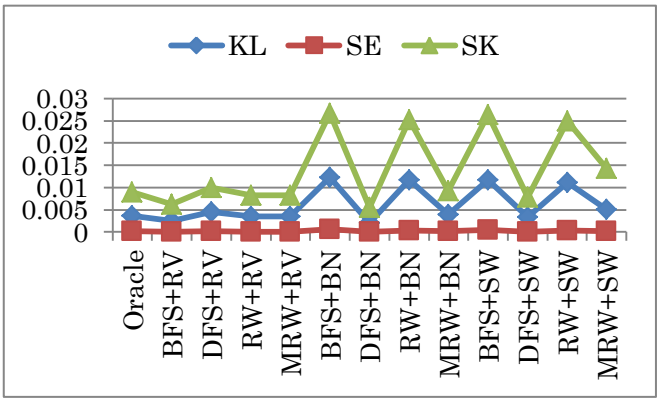

図 5 ランダムグラフ Uniform の評価結果(p=0.25)

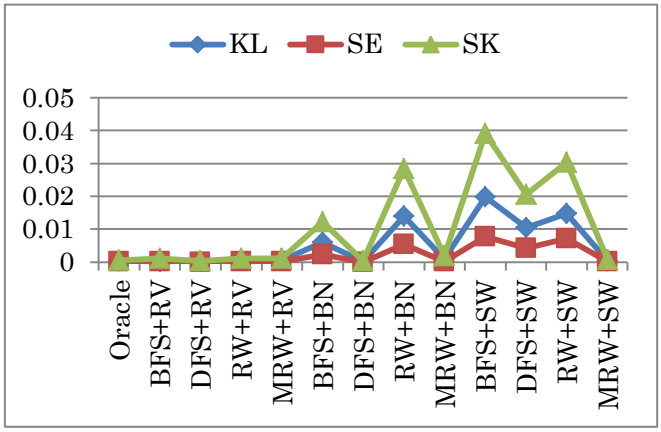

図 6 ランダムグラフ SmallWorld の評価結果 (p=0.25)

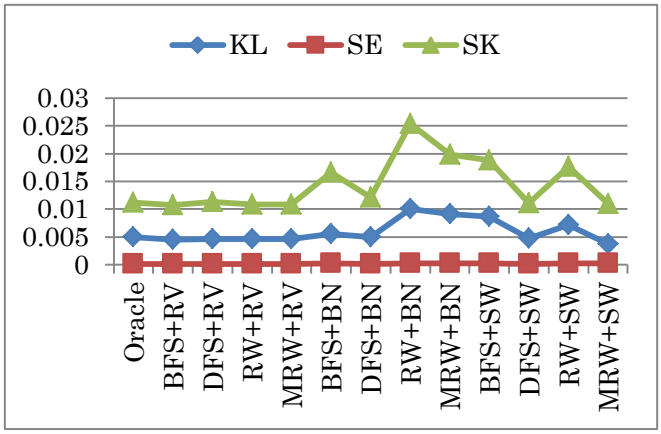

図 7 ランダムグラフ PowerLaw の評価結果(p=0.25)

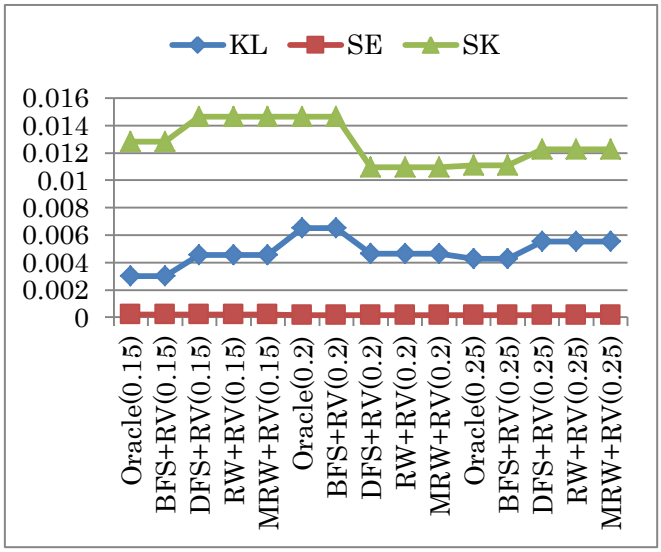

図 8 異なる抽出率の評価結果（RV の場合）

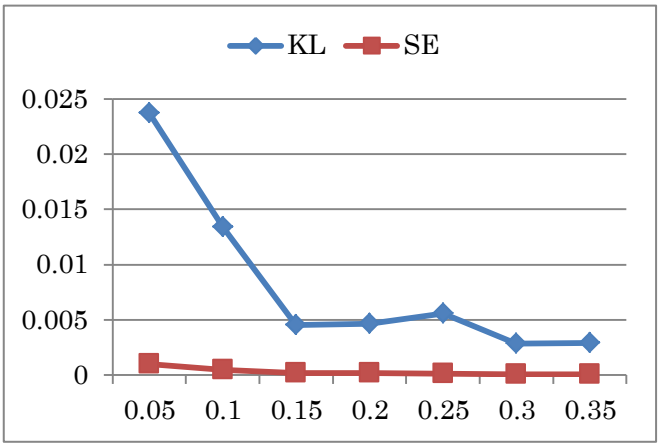

図 9 異なる抽出率の評価結果（BFS+RV）

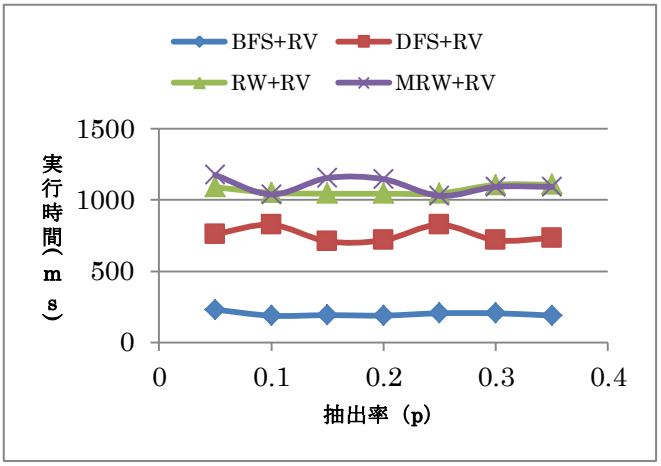

図 10 実行時間の評価結果（Reservoir）

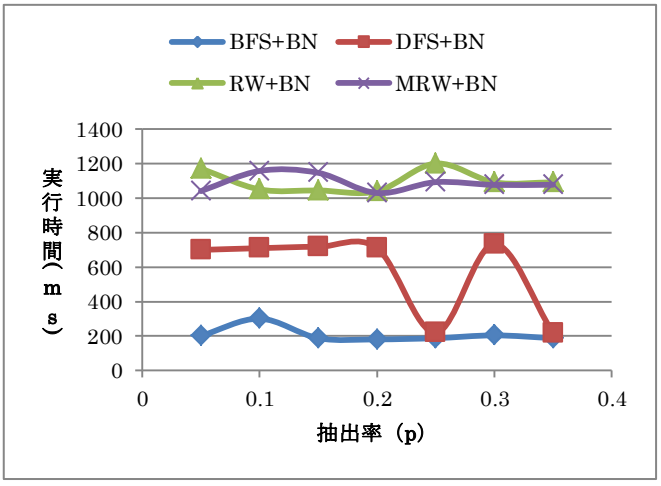

図 11 実行時間の評価結果（Bernoulli）

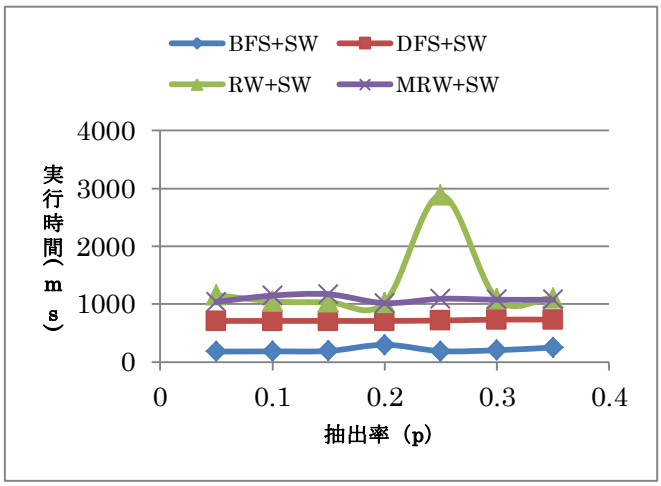

図 12 実行時間の評価結果（SlidingWindow）

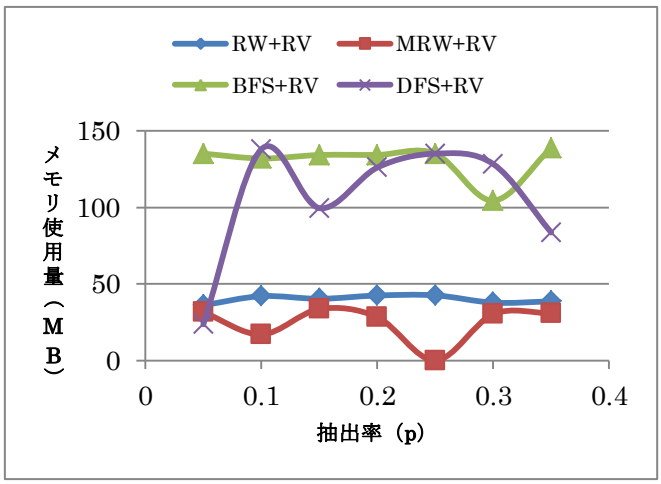

図 13 メモリ使用量の評価結果（Reservoir）

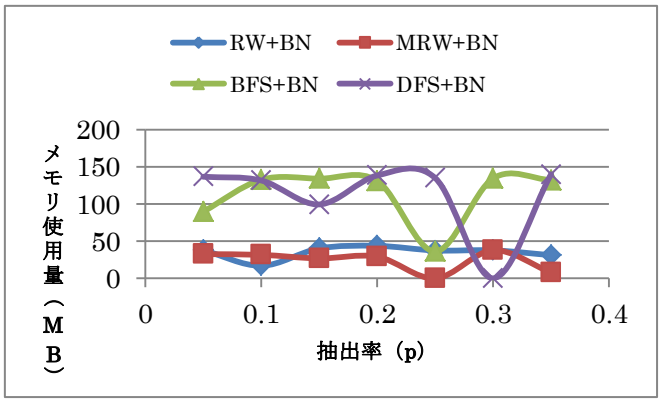

図 14 メモリ使用量の評価結果（Bernoulli）

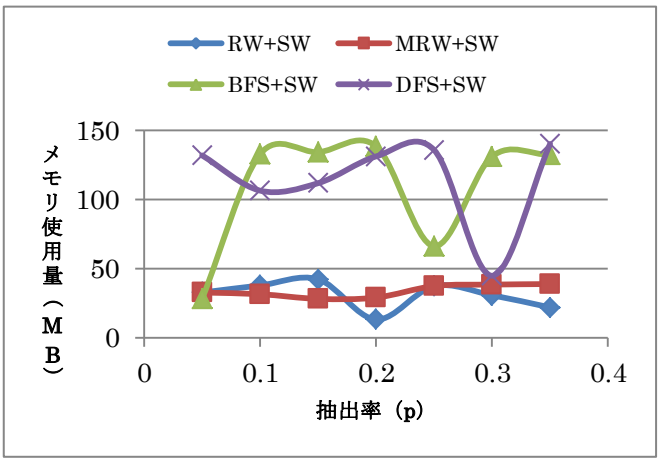

図 15 メモリ使用量の評価結果（SlidingWindow）

5.4. 考察

実験結果を見ると分かるように、Reservoir を用いる場合は、探索手法に影響を受けにくく、常によいサンプルを抽出ことができる。ほかの更新手法を使うとき、探索アルゴリズムは DFS と MRW を用いた結果がよかった。

RW は最も悪かった原因は、次数の高い頂点は訪問しやすく偏った結果になってしまうことである。しかし、RW はメモリ使用量が少なく、大規模なグラフに適している。

また、抽出率が高くなると、サンプルの質がよくなる傾向がある。

実行時間は抽出率とあまり相関関係を確認できていない。平均的に SW、RV、BN の順に時間効率がよい。BN は一番時間がかかる。

メモリ使用量を調べると、予想どおり RW、MRW のような未展開頂点を記憶する必要のない探索アルゴリズムのほうはよい。

6. 終わりに

本研究では適切なグラフサンプリング手法を設計するために、グラフ探索、サンプル抽出、サンプル更新を含む新しい枠組みを考案し、探索アルゴリズムやサンプル更新手法の重要性を示した。予備実験による評価を行った。

今回の評価実験は、機器設備の条件の制限で比較的に小規模のランダムグラフを使って行った。実データを使った大規模な実験は必要と思われる。また、探索アルゴリズムとグラフのカバー時間、カバー率の関係を調べる必要がある。

参考文献

[1] D. J. Aldous, Lower bounds for covering times for reversible Markov chains and random walks on graphs, J. Theoretical Probability 2(1989), 91–100

[2] M. Henzinger, A. Heydon, M. Mitzenmacher and M. Najork, On near-uniform URL sampling. In Proceedings of the 9th International World Wide Web Conference、2001，pp.295-308.

[3] K. Bharat and A. Broder, A technique for measuring the relative size and overlap of public Web search engines. In: Proc. of the 7th International World Wide Web Conference, Brisbane Australia, Elsevier, Amsterdam, 1998, pp. 379-388.

[4] D. Chakrabarti, C. Faloutsos; Graph mining: laws, tools, and case studies. Morgan & Claypool Publishers, 2012

[5] Y. Tille; Sampling algorithms, Springer, 2010

[6] S. Brin and L. Page, The anatomy of a large-scale hyper-textual Web search engine, in: Proc. of the 7th International World Wide Web Conference, Brisbane Australia, Elsevier, Amsterdam, 1998, pp.107-117

[7] Vitter,J.S; Random Sampling with a reservoir. ACM Trans. Math. Softw. 11(1) .1985 Mar.,pp.37-57

[8] W. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika, 57, 1(1970), pp.91-109.

[9] J Leskovec, C Faloutsos Sampling from Large Graphs. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006

[10] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. arXiv.org:0810.1355, 2008

[11] M. Henzinger, A. Heydon, M. Mitzenmacher and M. Najork, Measuring index quality using random walks on the Web. In Proceedings of the 8th International World Wide Web Conference(WWW8) 1999.pp 213-225

[12] Z.A.Bar-Yossef, A. Berg. S. Chin, and J. Fakcharoeenphol; Approximating aggregate queries about web pages via random walks. In Proceedings of the 26th International Conference on Very Large Data Bases. 2000

[13] P. Rusmevichientong, DM.Pennock, S. Lawrence, C. Lee Giles; Methods for sampling pages uniformly from the World Wide Web. In Proceedings of the AAAI Fall Symposium on Using Uncertainty Within Computation, 2001,Cape Cod, MA, pp.121-128

[14] 仲前　晋太郎,成　凱; Reservoir を用いた巨大グラフのランダムサンプリング, DEIM 2011

[15] D. Stutzbach,R. Rejaie, N. Duffield, S. Sen, and W. Willinger; On Unbiased Sampling for Unstructured Peer-to-Peer Networks. TON,16(2),Apr. 2008.

[16] A.H. Rasti, M. Torkjazi, R. Rejaie, N. Duffield, W. Willinger, D. Stutzbach; Respondent-Driven Sampling for Characterizing Unstructured Overlays. INFOCOM 2009, IEEE.

[17] 鹿島 久嗣, グラフとネットワークの構造データマイニング, 電子情報通信学会誌 93(9):797-802, 2010-09-01

[18] JUNG : Java Universal Network/Graph Framework、http://jung.sourceforge.net/