Watts–Strogatz モデルに基づく大規模ランダムグラフの分散並列生成

神野 薫　江本 健斗

近年，SNS や Web グラフなどの大規模なグラフに対して，解析処理を行うプログラムの開発の需要が高まってきている．そのようなプログラムの性能評価にあたっては，特定の特徴をもった様々なノード数のグラフが多く必要となる．しかし，逐次プログラムによる大規模な性能評価用グラフの生成は，非常に時間がかかってしまい，また，メモリ不足の可能性もあり，望ましくない．これらの問題点を解決するために，大規模グラフ生成の分散並列化が望まれる．

本研究では，現実世界の大規模グラフに似た特徴を持つ，よく知られたグラフモデルの一つである Watts–Strogatz モデルに基づく生成の分散並列化を提案する．この際，分散並列環境として Hadoop MapReduce を用いる．分散並列化にあたっては，元の生成法に若干の制約を導入し，その効率化を図っている．そのため，提案手法が元のモデルに期待される特徴を保存していることも，理論的・実験的に証明する．

In recent years, there has been a growing demand for the development of programs that perform analysis on large-scale graphs such as SNS and Web graphs. In evaluating the performance of such programs, we need a number of input graphs with the desired number of nodes and specific features such as the small-world property. However, the generation of large-scale random graphs by sequential programs is very time-consuming, and may cause memory shortage. In order to solve this problem, distributed parallelization of large-scale graph generation is desired.

In this research, we propose distributed parallelization of the random-graph generation based on Watts-Strogatz model, which is one of well-known graph models that provide characteristics similar to large scale graphs in the real world. We implemented our proposing distributed parallel algorithm by using Hadoop MapReduce. In the distributed parallelization, some restrictions are introduced into the original generation method to improve its efficiency. We have shown that the restrictions do not break the the characteristics that the original model has.

1　はじめに

ソーシャルネットワークやタンパク質間相互作用に代表されるような，大量の「物」とその間の「関係」とからなる巨大なデータの解析が広く行われるようになってきた．このような巨大データは，「物」を頂点，「関係」を辺とした大規模なグラフであると捉えられる．そのようなグラフの解析を行うプログラムの開発や，その開発を支援するためのフレームワークの需要が高まっている．

大規模グラフを処理するプログラムやフレームワークの性能評価では，現実のグラフを対象としたベンチマークの他に，将来的な使用に対しての傾向を捉えるために，ランダム生成されたグラフを用いた評価が行われる．すなわち，現実のグラフと同様の特徴を持つ指定サイズのグラフを生成できるモデルを用い，様々なサイズのグラフに対する処理時間等の測定を行う．それにより，例えば将来的に処理対象グラフのサイズが大きくなったときにどの程度の計算時間となるのかの知見が得られる．

大規模グラフ処理の評価に用いるべきランダム生成グラフは，当然ながら，大規模でなければ意味がない．しかし，大規模なランダムグラフの生成を 1 台の計算機で行うことは計算時間やメモリの制約から

Distributed parallel generation of large-scale random graphs based on Watts–Strogatz model.
Kaoru Kamino, Kento Emoto, 九州工業大学, Kyushu institute of technology.
コンピュータソフトウェア, Vol.37, No.2 (2020), pp.34–45.
[研究論文] 2019 年 8 月 5 日受付.

非現実的であり，ランダム生成モデル (アルゴリズム) の分散並列化が必要となる．

本研究は，様々なグラフ処理フレームワーク等の評価によく用いられる，Watts–Strogatz モデル [11] の分散並列化を提案する．WS モデルは，円状に並べられた n 個の頂点の各々が近隣の $2k$ 頂点と接続するリング状格子を初期グラフとし，各辺を確率 p で未接続頂点に再接続させることでグラフを生成する．このモデルは，各頂点に自身に接続する辺の半分の再接続を実行させることで容易に並列化できそうに思えるが，単純な並列化では再接続辺の重複の問題が生じてしまう．本研究は，この問題を再接続先頂点の選択に単純な制限を加えることで回避し，全体として頂点数程度の潜在的並列性を持つ生成アルゴリズムを提案する．本研究の貢献は次のとおりである．

- Watts–Strogatz モデル [11] に対し，潜在的並列度の高い分散並列化を提案する．本手法は，再接続辺重複問題を回避するために，辺の再接続先に簡単な制約を追加する．この制約により，各頂点が自身に接続する辺を独立にかつ安全に再接続処理することが可能となり，単純なデータ並列の枠組みを用いてグラフの分散並列生成が可能となる．
- 提案手法で追加する再接続先の制約が，Watts–Strogatz モデルに期待される諸性質を破壊しないことを確認した．生成アルゴリズムの安易な改変は，生成されるグラフに期待される諸性質を破壊してしまい，望ましくない．本研究では，Watts–Strogatz モデルに関する既存の解析 [2] に基づき，そこで解析された諸性質が提案手法を用いた場合にも保たれることを示した．
- 提案する Watts–Strogatz モデルの分散並列化を Hadoop MapReduce [10] を用いて実装し，元の生成アルゴリズムでは計算機一台では生成するのが困難であるような，大規模なグラフの生成が現実的に行えることを示した．

本論文の以降の構成は次のとおりである．まず，第 2 節にて，本研究で対象とする Watts–Strogatz モデルを導入する．次に，第 3 節にて，Watts–Strogatz モデルの並列化における問題点である再接続辺の重複を指摘するとともに，それを回避する分散並列化手法を提案する．続く第 4 節にて，提案手法が元の Watts–Strogatz モデルに期待される諸性質を保つことを示す．そして，第 5 節にて，提案手法の Hadoop Mapreduce による実装を導入し，PC クラスタにおける実験結果を示すとともに提案手法を評価する．最後に，第 6 節にて関連研究を述べ，第 7 節にて本論文をまとめる．

2 Watts–Strogatz モデル

本節は，一般によく使われるランダムグラフの生成モデルである Watts–Strogatz モデル [11] を導入する．このモデルは，単純 †1 な無向グラフを生成する．

Watts–Strogatz モデルで生成されるグラフは，平均経路長 (全頂点間距離の平均) が短いという「スモールワールド性」を持つことや，高めのクラスタ係数 (グラフ内の辺が三角形を構成している割合) を持つことといった，現実のグラフとよく似た特徴を示す．そのため，現実のグラフを対象としたいグラフ解析プログラム等の評価によく用いられる．

2.1 生成アルゴリズム

以下，Watts–Strogatz モデルでのグラフ生成の流れを説明する．生成は大きく二段階になっており，最初に規則的なリング状格子を形成し，その後，辺の確率的な再接続を行ってその規則性を乱す．

Watts–Strogatz モデルは三つのパラメータを持つ：

- N ··· グラフの頂点数
- K ··· 平均次数の半分 (すなわち，各頂点は平均で $2K$ 本の辺を持つ)
- P ··· 辺の再接続確率 (すなわち，グラフの乱れ具合)

2.1.1 リング状格子の形成

まず，N 個の頂点を円状に配置する．その後，各頂点が片側 K 個ずつ，合計 $2K$ 個の隣接頂点と接続するよう辺を追加する．こうして得られるリング状の規則的な格子 (リング状格子) が Watts–Strogatz モデルの初期グラフとなる．

†1 各頂点間に，辺は高々 1 本しかない．

例として，N=12，K=2 としたリング状格子を図 1 に示す．

このリング状格子は，隣接する頂点が三角形を多く形成しているため，現実のグラフと同様にクラスタ係数が高い．しかし，反面，頂点間の平均的な距離は長く，N/K に比例する程度である．そのため，次の段階として辺の確率的な再接続を行い，幾らかの「ショートカット」を導入することで平均経路長を小さくする．

2.1.2 辺の再接続処理

第二段階の処理として，すべての辺について確率的にその接続先を変更する再接続処理を行う．

再接続処理では，各頂点の各辺について順に，確率 P で接続先の変更を行う．この際，再接続先の頂点は，その辺が起点としている頂点が接続したことのない頂点から選択する．

擬似コードとして表現した再接続処理は次のとおりである．ここで，全頂点は $V=\{1,\ldots,N\}$ であり，v の担当する再接続処理対象の辺 (すなわち v を起点とする辺) は $E(v)=\{(v,v+1),(v,v+2),\ldots,(v,v+K)\}$[†2] である．すなわち，各頂点は自身と自身以降の頂点を結ぶ K 本の辺を担当する．また，rand() を $[0,1]$ の一様ランダムな値を返す関数とする．

foreach $v \in V$
  foreach $e \in E(v)$
    if rand() $> P$ **then**
      e の終点を V' から一様ランダムに選択
        where
          $V' = V \setminus \{v-K,\ldots,v,\ldots,v+K\}$
                $\setminus$ { 再接続で v と繋がった頂点 }

図 1 のグラフに対して再接続処理を行った例を図 2 に示す．頂点 1 から頂点 3 に伸びていた辺に対して接続先の変更が起こり，その辺の終点が頂点 5 に変更された．同様に，頂点 8 から頂点 9 に伸びていた辺に対しても接続先の変更が起こり，その辺の終点が頂点 12 に変更された．いずれの辺についても，再接続先はそれぞれの起点であった頂点が接続したことのなかった頂点であることが大事である．これにより，「辺が重複して本数が減る」という事態が避けられている．

†2 簡潔な記述のため，$N+1$ と 1 を同一視するなど，周期 N での同一視を行うとする．

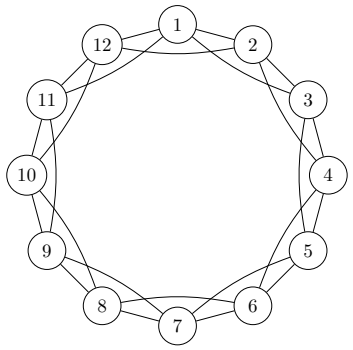

図 1 リング状格子の例 ($N=12$，$K=2$)．頂点内の数は，各頂点の番号である．

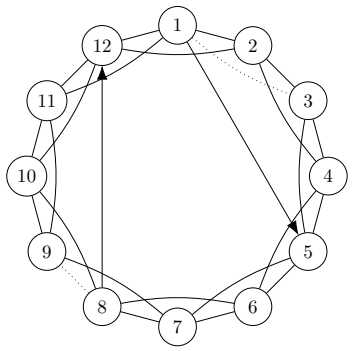

図 2 再接続処理の例：頂点 **1** と頂点 **8** の辺 (点線) が再接続 (矢印) を行った．

3 Watts–Strogatz モデルの分散並列化

本節では，Watts–Strogatz モデルの単純な並列化の直面する問題点を指摘した後に，その問題を解決した Watts–Strogatz モデルの分散並列化を提案する．

3.1 単純な並列化とその問題点

前節で述べた Watts–Strogatz モデルの並列化は，一見，全頂点での再接続処理を並列に行えば簡単に実現できるように見える．しかし，この単純な並列化では「再接続辺の重複」の問題が生じる．

図 1 のリング状格子に対して再接続処理することを考える．ここで，全頂点が独立に並列に，その担当

する辺の再接続処理を行ったとしよう．このとき，図 3 に示すような状況が生じる．すなわち，頂点 1 が頂点 3 に繋がっていた辺を頂点 8 に再接続し，同時に，頂点 8 が頂点 9 に繋がっていた辺を頂点 1 に再接続するという状況が起こり得る．すなわち，再接続辺が重複してしまうことが起こり得る．

上記の再接続辺の重複は，いくつかの問題を生じる．まず，生成されるグラフは単純であるとしているため，重複した二本の辺は一本に同一視され，グラフ全体の辺の数が期待された数より小さくなってしまう．また，グラフが単純であると想定しているプログラムに上記の重複辺をそのまま入力してしまえば，思わぬ不正処理を発生させる事になりかねない．そして，既存の理論的な解析は重複辺の存在を仮定していないため，Watts–Strogatz モデルに期待される性質を破壊してしまう恐れもある．

以上のような理由により，各頂点を独立に並列に動かすという単純な並列化は望ましくなく，重複辺の発生を防ぐ仕組みを導入した並列化が望ましい．ひとつの方法として，「再接続時にロックを取る」などの排他制御を入れることも考えられるが，並列計算に用いる計算機が多い場合にはあまり現実的ではない．また，「重複した際にやり直しをする」という手法も考えられるが，重複の有無の確認や再実行のコストが掛かるという問題がある．

本研究は，「重複辺が生じない選択しかできないようにする」という手法による解決を次節で提案する．

3.2 提案手法：重複回避を伴う並列化

本研究は，前節で指摘した再接続辺重複の問題を，「重複辺を生成しない再接続先の選択方法」を導入することにより解決する．

再接続辺の重複が起こるのは，二つの頂点が，互いに互いを再接続先に選んでしまったときである．よって，再接続先頂点の選択に非対称性を導入し，「一方から他方は選べるが，その逆は不可能である」という状況を作れば良い．

上記のような非対称的な選択を行うひとつの手法として，各頂点 v における再接続先を次のようにすることを提案する：

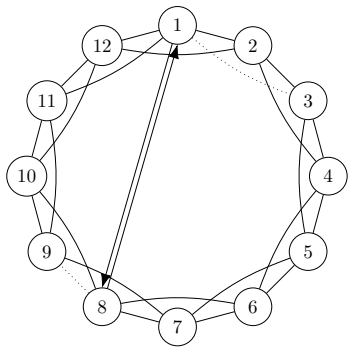

図 3 再接続処理の並列化時に発生し得る辺の重複：頂点 1 と頂点 5 が，互いを再接続先に選んでしまった状況.

v から円の中心を見て，左側の半円の頂点のうち v と頂点番号の偶奇が一致する頂点と，同じく右半円のうち偶奇の異なる頂点.

すなわち，第 2 節に示した擬似コードにおいて，頂点 v の再接続先候補集合 V' を次のようにすることを提案する．

$$
\begin{aligned}
V' = {} & \{v + K + i \mid i \in \{1, 2, \ldots, \lfloor (N-1)/2 \rfloor - K\}, \\
& K + i \text{が偶数} \} \\
& \cup \{v - K - i \mid i \in \{1, 2, \ldots, \lfloor (N-1)/2 \rfloor - K\}, \\
& \quad K + i \text{が奇数} \} \\
& \setminus \left\{ v \text{の再接続で} v \text{と繋がった頂点} \right\}
\end{aligned}
$$

なお，N が偶数である場合，真反対の頂点とは接続できないものとする．

再接続先候補に上記の偶奇制約を課すことで重複辺の生成を回避できることは自明である．最初のリング状格子で接続されていなかった頂点 u と v について考える．u が v の右半円に存在すると仮定しても一般性は失われない．このとき，v は u の左半円に存在することに注意する．u と v の偶奇が同じである場合，u から v に向けての再接続は可能であるが，逆に，v から u への再接続は上記の制約によって起こり得ない．同様に，u と v の偶奇が異なる場合，v から u に向けての再接続は可能であるが，逆に，u から v への再接続は上記の制約によって起こり得ない．いずれにせよ，u から v へと v から u へとの両方の再接続は同時には起こり得ない．

図 4 と図 5 に，N=12，K=2 とした時の，頂点 1

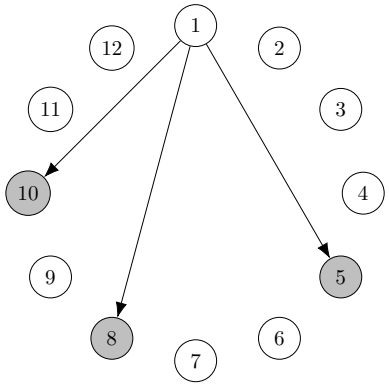

図 4　頂点 1 から接続可能な頂点

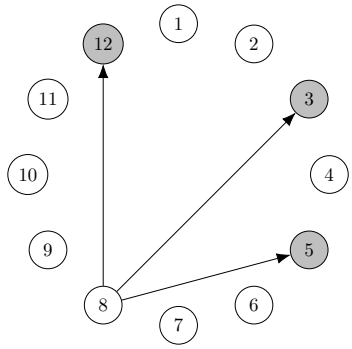

図 5　頂点 8 から接続可能な頂点

と頂点 8 に対する選択可能な再接続先頂点の様子を示す．まず，頂点 1 について，円の中心を見た際の左半円にある頂点は 2，3，4，5，6 である．これらのうち，実際に再接続先として選択可能なのは，最初のリング状格子で接続されておらず且つ偶奇の一致する頂点 5 のみとなる．同様に，右半円には 8，9，10，11，12 の頂点があるが，これらのうち，最初のリング状格子で接続されておらず且つ偶奇の異なる頂点 8 と頂点 10 が接続先候補となる．総じて，頂点 1 の再接続先候補の頂点は 5，8，10 の三つである．他方で，頂点 8 の再接続先候補の頂点は，3，5, 12 となる．そのため，頂点 1 と頂点 8 の間に再接続辺が生じ得るとすれば，それは頂点 1 にしか行うことができず，前節で指摘した再接続辺重複の問題は生じない．

以上の「重複辺を生成しない再接続先の選択方法」の導入により，グラフの生成を安全に且つ単純に，頂点毎に独立な計算として並列化することが可能となった．すなわち，第 2 節の擬似コードの最外の **foreach** を，単純なデータ並列の形で分散並列化することが可能となった．

4　諸性質の保存

前節で提案した「再接続先頂点選択への制約導入」の手法により，グラフ生成の安全且つ単純な分散並列化が実現できる．しかし，この制約の導入によって，生成されるグラフに期待されていた性質が壊されてしまうようでは本末転倒である．

本節では，前節の提案手法が，Watts–Strogatz モデルで生成されるグラフに期待される性質を保っていることを示す．具体的には，Watts–Strogatz モデルの解析を行いその性質を整理した研究 [2] に倣い，本提案手法で改変した生成方法について同様の解析を行う．

4.1　接続性の分布の保存

本節では，グラフの接続性，すなわち，頂点次数の分布の保存の確認を行う．そのためにまず，元の Watts–Strogatz モデルで生成した場合の分布を計算し，その後，提案手法で改変された場合の分布を計算する．

以下，生成されたグラフにおいて，ある頂点が d 本の辺 (隣接頂点) を持つ確率 $P(d)$ を計算する．そのためにまず，頂点の持つ辺を種類分けして整理する．

生成されたグラフにおいて，頂点の持つ d 本の辺は，大きく二種類に分類される．一つは自身が再接続を行う K 本の辺であり，常に K 本存在する．すなわち, 必ず $d \geq K$ である．もう一方は，他の頂点の再接続処理により変わる部分であり，$n_i = d - K$ 本存在する．後者はさらに，リング状格子の段階で接続されていた辺 $n_{i,1}$ 本 (最大 K 本) と，別の頂点の再接続処理によって繋げられた辺 $n_{i,2} = n_i - n_{i,1}$ 本とに分けられる．

ここから，確率 $P(d)$ の具体形を求めていく．ここで，自身が再接続を行う K 本の辺は，確率に依存せず常に K 本存在することに注意する．よって，$P(d)$ は，$n_i = d - K$ の $n_{i,1}$ と $n_{i,2}$ への割り振りの全パタンを足し合わせたものとなり，前者に関する確率を $P_1(n_{i,1})$，

後者に関する確率を $P_2(n_{i,2}) = P_2(d-K-n_{i,1})$ と書けば，$d \geq K$ に対して次式となる．

$$P_p(d) = \sum_{n_{i,1}=0}^{min(d-K,K)} P_1(n_{i,1})P_2(d-K-n_{i,1})$$

リング状格子の段階で接続されていた辺が $n_{i,1}$ 本残っている確率 $P_1(n_{i,1})$ は，次式で与えられる．

$$P_1(n_{i,1}) = \binom{K}{n_{i,1}} (1-P)^{n_{i,1}} P^{K-n_{i,1}}$$

すなわち，$n_{i,1}$ 本のそれぞれが $1-P$ の確率で取り残された確率である．

他方，再接続処理により新しく v に接続されることとなった辺が $n_{i,2}$ 本となる確率は，$n_{i,2}$ 個の頂点が再接続先として v を選ぶという確率のことである．そのような頂点の各々が v を再接続先に選ぶ確率は，$N \gg K$ である場合 †3，KP/N である．すなわち，そのような頂点は K 回の再接続のチャンスが有り，各々の再接続は確率 P で成功するもので，さらに，およそ N 個の再接続先から v を選ぶ確率は $1/N$ である．よって，そのような頂点が全体でおよそ N 個あることに注意すれば，次式を得る．

$$P_2(n_{i,2}) = \binom{N}{n_{i,2}} \left(\frac{KP}{N}\right)^{n_{i,2}} \left(1-\frac{KP}{N}\right)^{N-n_{i,2}}$$

さらに，既存の解析 [2] と同様に N を大きくした際の極限を取れば，上記の二項分布が $\lambda = N \times (KP/N) = KP$ のポアソン分布になり (例えば文献 [8] などを参照)，次式を得る．

$$P_2(n_{i,2}) = \frac{(KP)^{n_{i,2}}}{n_{i,2}!} e^{-KP}$$

よって，以上から，元の Watts–Strogatz モデルでの生成における $P(d)$ は，次式となる．ただし，$d \geq K$ であることに注意する．

$$P_p(d) = \sum_{n=0}^{min(d-K,K)} \binom{K}{n} (1-P)^n P^{K-n} \times \frac{(KP)^{d-K-n}}{(d-K-n)!} e^{-KP}$$

次に，本研究の提案手法で改変した生成法に関する $P(d)$ を計算する．変更される点は，P_2 の計算における以下の二点である．まず，各頂点が v に接続する確率が倍の $2KP/N$ になる．これは，再接続先の選択肢が，導入された偶奇制約により半分になるからである．次に，v に再接続することのできる頂点の数が，同様の理由により半分の $N/2$ になる．よって，提案手法における $n_{i,2}$ に関する確率 $P_2'(n_{i,2})$ は，次式となる．

$$P_2'(n_{i,2}) = \binom{\frac{N}{2}}{n_{i,2}} \left(\frac{2KP}{N}\right)^{n_{i,2}} \left(1-\frac{2KP}{N}\right)^{\frac{N}{2}-n_{i,2}}$$

ここで，P_2 と同様に $N \to \infty$ の極限を取ってポアソン分布に近似することを考えると，$\lambda = N/2 \times 2KP/N = KP$ となり，結果は $P_2'(n_{i,2}) = P_2(n_{i,2})$ となる．導入した制約により影響を受け得るのは P_2 のみであるため，$P(d)$ も一致する．

以上により，提案手法によるグラフ生成は，元の Watts–Strogatz モデルでの接続性を保存することが確認された．

4.2 クラスタ係数の保存

次に，クラスタ係数の保存を確かめる．前節と同様，まずは既存の解析 [2] に従い元の Watts–Strogatz モデルでのクラスタ係数を求め，その後，提案手法による生成での結果について述べる．なお，既存の解析では真のクラスタ係数ではなくその近似値を用いて解析を行っている．よって，本論文でも同じ近似値での比較を行う．

グラフのクラスタ係数 C は，そのグラフ内の三角形の割合のことであり，各頂点のクラスタ係数 c_v の平均値で定義される：

$$C = \mathrm{E}(c_v)$$

頂点 v について，その隣接頂点数を d_v，その隣接頂点間に存在する辺の本数を N_v とすると，頂点 v のクラスタ係数 c_v は次式で定義される：

$$c_v = \frac{N_v}{d_v(d_v-1)/2}$$

すなわち，v の周辺について，存在し得る三角形の最大数 $d_v(d_v-1)/2$ と，実際に存在している三角形の数 N_v との比である．

まず，$P=0$ のときを考える．すなわち，再接続

†3 既存研究 [2] と同様の仮定

が行われずリング状格子が生成結果となる場合を考える．このとき，v の左右には K 個ずつの隣接点がある．この隣接する頂点間にある辺の数 N_v は，頂点番号の低いほうを起点に辺を数え上げるとすると，次のようになる．まず，v より小さい K 個の頂点の各々を考えると，それらは v を抜いた $K-1$ 個の隣接頂点と接続している．よって，これらをまとめると $K(K-1)$ 本の辺が存在する．次に，v より大きな頂点については，v の真横の頂点が $K-1$ 本，その隣が $K-2$ 本，……，などのように辺を持っている．これらをまとめれば，$K(K-1)/2$ 本の辺が存在する．よって，これらの和を $\bar{N}$ として，N_v は次式となる．

$$N_v = \bar{N} = K(K-1) + \frac{K(K-1)}{2} = \frac{3K(K-1)}{2}$$

さらに，このとき，$d_v = 2K$ であるため，このときのクラスタ係数 $C(0)$ は次式となる．

$$\begin{aligned} C(0) &= \mathrm{E}\left(\frac{N_v}{c_v(c_v-1)/2}\right) \\ &= \mathrm{E}\left(\frac{3K(K-1)/2}{2K(2K-1)/2}\right) \\ &= \frac{3(K-1)}{2(2K-1)} \end{aligned}$$

次に，$P > 0$ の場合のクラスタ係数 $C(P)$ を考える．厳密に $C(P)$ を求めることは困難なため，既存の解析 [2] では期待値と演算の入れ替えを行った近似値による解析を行っている．以下，それに倣った解析を行う．

$P > 0$ の場合の N_v の期待値を計算する．そのために，初期状態であるリング状格子 (すなわち $P = 0$ の状態) から再接続処理を行い，三角形の数がどのように変化するかを考える．まず，既存の三角形が保存される確率は，その三角形を構成している三本の辺全てが再接続されない確率に等しい．よって，保存される三角形の個数の期待値は，$\bar{N}(1-P)^3$ である．次に，新しく三角形が構成される確率を考える．新たな三角形が構成されるには，v とは隣接していなかった 2 頂点を考え，それら 3 点が互いに互いを接続先として再接続を行わなければならない．その確率は，$\mathrm{O}(N^2) \times \mathrm{O}((1/N)^3) = \mathrm{O}(1/N)$ である．また，v の隣接頂点と初期の辺を新たな三角形の一部とする可能性もあるが，その場合も同様に $\mathrm{O}(1/N)$ である．よって, $P > 0$ の時の N_v の期待値は，次式となる．

$$\mathrm{E}(N_v) = \bar{N}(1-P)^3 + \mathrm{O}\left(\frac{1}{N}\right)$$

ここで，$\mathrm{O}(1/N)$ の項は N が十分に大きい時には無視できる．従って，$P > 0$ の時のクラスタ係数の近似値 $\tilde{C}(P)$ が次式となる．

$$\begin{aligned} \tilde{C}(P) &= \frac{\mathrm{E}(N_v)}{\mathrm{E}(c_v(c_v-1))/2} \\ &= \frac{\bar{N}(1-P)^3}{\mathrm{E}(c_v(c_v-1))/2} \end{aligned}$$

さらに，分母についても平均と演算の順序を入れ替えてしまえば，$\mathrm{E}(c_i) = 2K$ より，最終的に次式を得る．

$$\begin{aligned} \tilde{C}(P) &\sim \frac{\bar{N}(1-P)^3}{\mathrm{E}(c_v)\mathrm{E}(c_v-1)/2} \\ &= \frac{\bar{N}(1-P)^3}{2K(2K-1)/2} \\ &= C(0)(1-P)^3 \end{aligned}$$

以上が，元の Watts–Strogatz モデルで期待されるクラスタ係数である．

次に，本研究で提案する手法について同様の値を計算する．導入した制約により変化し得る部分は，再接続により新たな三角形が構成される部分である．しかし，上記の解析において，新たな三角形を構成しうる (v 以外の)2 頂点の選択の幅が半分になろうとも，再接続により互いが接続され得る確率が倍になろうとも，新たな三角形が構成される可能性が $\mathrm{O}(1/N)$ であることに変わりはない．よって，提案手法に関する解析も，上記の元の Watts–Strogatz モデルに関する解析と同じ結果を得る．

以上により，提案手法が元の Watts–Strogatz モデルに期待されるクラスタ係数の特性を保存することが確認された．

4.3 平均経路長の分布の保存

平均経路長とは，ノード間距離 (経路長) のグラフ全体での平均の値のことである．Watts–Strogatz モデルの平均経路長の分布を理論的に解析することは困難であるため，既存の解析 [2] では，理論的な解析を諦めてその分布を実験的に測定して解析を行って

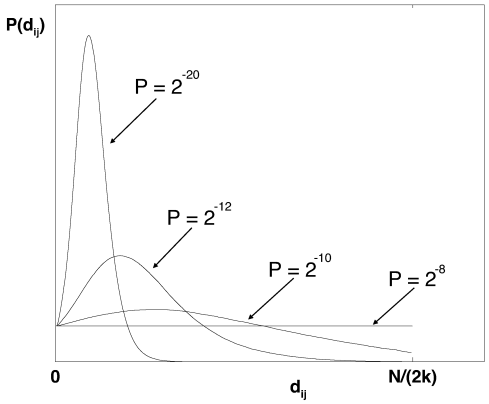

図 6 ノード間距離の割合 (文献 [2] から引用)

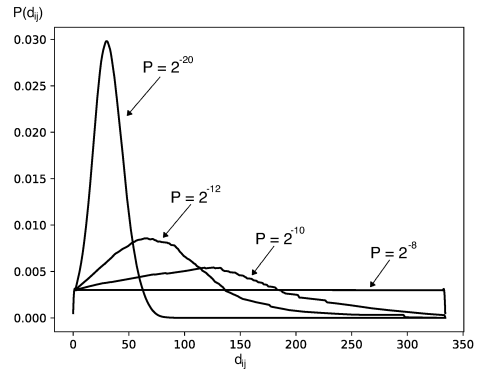

図 7 ノード間距離の割合 (提案手法)

いる．本研究でもその手法を踏襲する．

既存の解析 [2] における平均経路長の分布の様子を図 6 に示した．この図は，$N = 2000$，$K = 3$ とし，P の値を 2^{-20}, 2^{-12}, 2^{-10}, 2^{-8} と変化させ，複数のグラフについて経路長ごとの割合を測定しプロットしたものである．P の値が小さいほどグラフが平坦であり，P が大きいほど山なりの部分が大きい．経路長の最大は $2K/N$ であり，P の値が大きくなるほど平均経路長が短くなる傾向がある．

図 7 に，同じパラメータに対して我々の提案手法で生成したグラフに関しての平均経路長をプロットした図を示す．この図は，元の生成法に関する図 6 と同様の特徴を示している．

以上のことから，本研究で提案した手法が平均経路長の分布に関する特徴も保存していることが確認された．

5 実装と評価

本節では，第 3 節で提案した手法の実装について簡単に述べ，その後，その実装を用いた提案手法の評価について述べる．

5.1 提案手法の実装

提案手法を大きく二つの枠組みで実装した．

5.1.1 Python による実装

一つ目の実装は Python による逐次実装である．その主な目的は，元の Watts–Strogatz モデルに基づく既存のランダムグラフ生成ルーチンと，提案手法に基づく生成ルーチンとの性能比較である．

グラフ処理のために広く利用されているフレームワークの一つに，Python で実装された NetworkX [9] がある．提案手法と既存手法のフェアな比較のため，同一言語での実装を行った．

なお，第 2 節のモデルの説明では初期のリング状格子を明示的に作成するとしていたが，実際の実装ではその必要がないことに注意する．すなわち，各頂点 v は，$2K$ 本の辺の分だけ $[0, 1]$ 区間の一様乱数の生成を行い，各値が P 以下のときにはリング状格子での隣接点をその辺の接続先に設定し，そうでなければ再接続先頂点の選択ルールに従ってランダムに選ばれた頂点をその辺の接続先に設定する．

また，我々の提案手法の実装では，辺の再接続処理を逐次的に各頂点独立に行った後，改めて隣接リストの双方向化を行っている．すなわち，頂点の u の持つ辺 (u, v) について，その接続先の頂点 v の隣接リストに u を入れるという処理を後で一斉に行う．

5.1.2 Hadoop MapReduce による実装

もう一つの実装は Hadoop MapReduce [4][10] を用いた分散並列実装である．こちらの実装言語は Java である．以降，その実装の概略を説明する．

Map タスクは，入力として，そのタスクで処理を担当する頂点番号の範囲 $[s, e)$ を受け取る．この他，モデルのパラメータ N, K, P を適切に受け取っておく．そして，各頂点 $v \in [s, e)$ に対して，Python のプログラムと同様，$2K$ 本の辺の接続先を計算する．

そして，それらの各々の接続先 w について，(v, w) と (w, v) という key-value ペアを生成する．

Reduce タスクは，生成されるグラフの各頂点 v 毎に存在する．先に説明した Map タスクの動きにより，v に接続する頂点のリストを受け取ることになる．そのため，そのリストを後のプログラムに使いやすい形に整形して出力するなどを行うのみとなる．

5.2 実験と評価

前節で説明した実装を用いて，提案手法の既存ルーチンとの性能比較や，分散並列実装時の性能を評価した．評価には，以下に示す環境の PC 16 台からなるクラスタを用いた．また，HDFS のチャンクサイズは 4MB とした．

CPU：Intel(R) Core(TM) i5 6500 @ 3.20GHz
メモリ：16GB (8GB x2, PC4 17000)
OS：Ubuntu 14.04.5 LTS
Java：Oracle JDK 1.8.0_131
Hadoop：1.2.1
Python：3.4.3

5.2.1 分散並列実行の評価

まず，$K = 4$, $P = 0.4$ と，$N = 100$ 万, 1000 万, 1 億に対して，それぞれの実装でグラフを生成するのにかかった時間を測定した．結果を表 1 に示す．なお，Hadoop 実装の Map タスクに与えられる入力の範囲は $[v, v + 1)$ とした．すなわち，一つの頂点について一つの Map タスクが生成される状況である．

まず，逐次実装である既存の NetworkX と提案手法の Python 実装を比較する．結果より，提案手法の方が 1.5 倍ほどの計算時間がかかることがわかる．これは，実装そのものが拙いことと，双方化の処理に時間がかかっていることによると考えられる．双方向化の処理を再接続処理時にまとめて行うことで，処理時間の差が縮まると考えられる．何れにせよ，提案手法による大幅な処理増加などがないことを確認できる．

次に，提案手法の Hadoop 実装と他の逐次実装とを比較する．

生成したいグラフサイズが小さい場合，Hadoop 実装で十分なタスク分割が行われず (Map タスクへの入力を記したファイルが分割されず)，逐次実装のほうが速く生成可能である．実際，100 万頂点では計算機代数よりも少ない数のタスクしか生成されず，計算機台数を増やしても計算時間は短縮されない．

他方, 生成するグラフサイズを大きくすると, Hadoop 実装の分散並列実行の効果が現れるようになる．実際，1000 万頂点でも，タスク数による頭打ちはあるものの計算機代数を増やすにつれて計算時間が短縮され，明らかに逐次実装よりも速くグラフ生成を終えている．さらに，1 億頂点では，もはや逐次実装ではメモリ不足等によりグラフの生成自体が不可能になっているのに対し，分散並列実装の Hadoop 実装では，1 億以下の頂点数については，例え計算機 1 台であろうとも生成を正常に終えることができた．これは，今回提案するアルゴリズムに基づく Hadoop 実装では中間結果を逐次的にディスクに吐き出すことが可能となり，必要とするメモリ量が小さくなっている事による結果である．なお，10 億頂点以上では，提案する Hadoop 実装でも 1，2 台での生成ができなかった．これは，Hadoop の reduce プロセスが，複数のキーに対する値を同時にメモリ上に保持しようとするため，メモリ不足に陥ってしまっているからである．また，Hadoop 実装では，投入する計算機台数に反比例する形で生成時間の短縮が達成できている．

以上の結果により，提案手法を用いることで，本研究の目的であった「一台の計算機では生成しにくいサイズでの Watts–Strogatz モデルのランダムグラフ生成」が達成されたことが確認された．

5.2.2 生成時間のモデルパラメータへの依存

次に，提案手法を用いた場合のグラフ生成時間の，モデルパラメータ N，K，P への依存を評価する．

Hadoop 実装について，計算機を 16 台用いた際の，各パラメータ N，K，P だけを変化させた場合の計算時間の変化を図 8 から図 10 に示す．

まず，図 8 より，生成時間が N に比例していることがわかる．また，図 9 より，生成時間が K にも比例していることがわかる．提案手法の計算量は明らかに $\mathrm{O}(NK)$ であるので，これは期待通りの結果である．

次に，P の値の生成時間への影響を見る．図 10 よ

表 1 ランダムグラフ生成時間 (秒)

頂点数	NetworkX (既存逐次実装)	Python 実装 (提案逐次実装)	Hadoop 実装 (提案並列実装)				
			1 台	2 台	4 台	8 台	16 台
100 万	6	9	19	17	16	16	17
1000 万	66	92	83	56	30	24	21
1 億	–	–	833	414	213	139	78
10 億	–	–			3811	1288	651

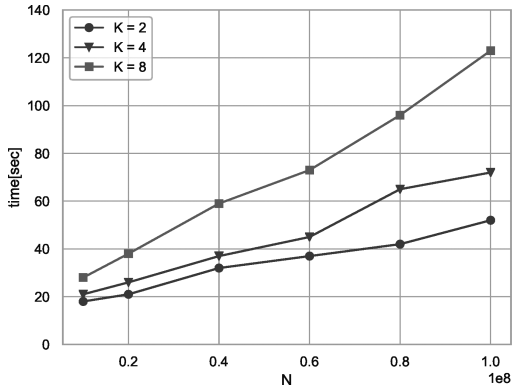

図 8 N だけを変化させたときの生成時間 ($P = 0.4$)

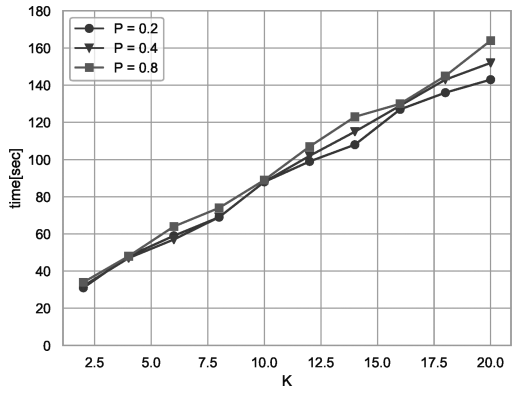

図 9 K だけを変化させたときの生成時間 ($N = 6000$ 万)

り，P の計算時間への影響は小さいが，P が大きくなるにつれ，多少時間が伸びる傾向が観察される．これは，P が大きくなると，単純なリング状格子の接続に比べて接続先の決定が複雑となる再接続処理が増えるためである．すなわち，予想される通りの結果である．

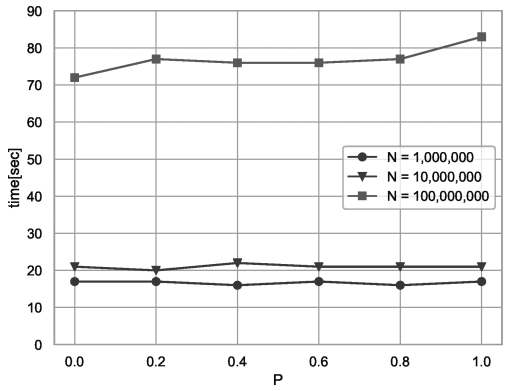

図 10 P だけを変化させたときの生成時間 ($K = 4$)

表 2 各ブロックサイズにおけるグラフ生成時間 (秒)

生成頂点数	ブロックサイズ				
	1	2	4	8	10
100 万	12	14	13	17	16
1000 万	17	14	17	18	22
1 億	82	62	48	47	55

5.3 Map タスクへの入力と計算時間

我々の Hadoop 実装は，Map タスクへの入力区間 $[s, e)$ のブロックサイズ $e-s$ に処理時間が左右される．以下，その影響を測定した結果を示す．

使用計算機台数を 16 台に固定し，$K = 2$, $P = 0.4$ とした際の，ブロックサイズに応じたグラフ生成時間の様子を表 2 に示す．

表 2 より，生成すべきグラフサイズが大きい場合には，ブロックサイズを大きくすることが望ましいとわかる．ただし，今回の実装では，入力とするブロックを 1 行毎並べたファイルを MapReduce への

入力としているため，ブロックサイズが大きくなるにつれて入力ファイルが小さくなり，十分な Map タスクの生成ができず並列化効率が低下してしまっている．適切な Map タスクの生成が望ましい．なお，入力ファイル自体の生成にも多少の時間がかかる．入力ファイルは N が変わる毎に必要となるが，ひとつの N に対して複数のグラフを生成するであろうことを考えると，その生成時間は無視しても良い．実際，例えば，1 億頂点における入力ファイル生成時間は，ブロックサイズが 1 の場合に 105 秒，ブロックサイズが 8 の場合に 13 秒程度である．

6 関連研究

Watts–Strogatz モデル以外にも様々なランダムグラフ生成モデルが知られている．ランダムグラフ生成モデルにより生成されるグラフには現実世界に見られるような性質を模倣できていることが期待される．現実のグラフが持つ性質として，例えば，本研究で用いた生成モデルである Watts–Strogatz モデルで生成されるグラフが持つ，スモールワールド性が挙げられる．その他，膨大なリンクを持つ頂点が少数存在するようなスケールフリー性なども挙げられる．ここでは, 関連研究としてこれらの性質を持つようなグラフを生成できるモデルを説明していく．

Erdős–Rényi モデル [5] は，最も単純なランダムグラフ生成モデルである．その生成法は単純で，各ノードが独立に，自分以外の各ノードに対し，与えられた確率 P で辺を張るというものである．言い換えれば，完全グラフの各辺について，確率 $1-P$ で削除を行うというものである．生成法の単純さ故に容易に分散並列化が可能であるが，しかし，現実のグラフに見られるようなスケールフリー性や高いクラスタ係数などの性質を持つグラフは生成されない．このモデルに基づくグラフの並列生成には，例えば GPU を用いた生成システム PPreZER [3] も存在する．

Barabáshi–Albert モデル [1] は，Watts–Strogatz モデルに並んで広く認知されているランダムグラフ生成モデルである．このモデルで生成されるグラフは，膨大なリンクをもつ頂点 (ハブ) が少数存在するような，スケールフリー性を備えることが知られている．そのため，現実のグラフの良い模倣として利用できる．このモデルでのグラフ生成は，基本的には頂点を一つずつ既存のグラフに追加していくことで行う．頂点の追加の際には，既に多くの辺を持っている頂点に接続しやすくするため，頂点次数に比例した確率で接続先の選択を行う．残念ながら，この強く逐次的な生成法は，並列化が容易ではない．

Klemm–Eguíluz モデルは [6]，Barabáshi–Albert モデルを元にしたモデルで，Watts–Strogatz モデルが持つスモールワールド性と Barabáshi–Albert モデルが持つスケールフリー性を両立するモデルである．その生成法の改変は，Barabáshi–Albert モデルにおける新しい頂点の追加の際に，古い頂点の中から以降の接続に関与しない頂点を作るというものである．これにより，Barabáshi–Albert モデルになかった性質を得られるものの，その生成法は依然として強く逐次的であるため，並列化は難しい．

Kronecker Graphs [7] は，クロネッカー積に基づいたランダムグラフ生成モデルであり，現実のグラフと同様の性質をもつグラフを生成できることが知られている．また，その生成の並列化も，Barabáshi–Albert モデル等に比べて容易である．この生成モデルは，例えば Graph500 の巨大な入力グラフの生成などに用いられている．

7 まとめ

本研究では, グラフ解析プログラム等の評価によく使われるランダムグラフ生成モデルである，Watts–Strogatz モデルの分散並列化手法を提案した．その基本的アイデアは，並列化の際に問題となる再接続辺の重複を避けるために，再接続先選択に簡単な偶奇制約を導入するというものである．また，その制約の導入によって元の Watts–Strogatz モデルの備えていた特徴を破壊していないことも確認した．さらに，提案手法を Hadoop MapReduce を用いて実装し，分散並列実行による生成時間の短縮と，一台の計算機では対応しにくいサイズのグラフの生成が容易となったことを確認した．

今後の課題として，提案手法のより効率的な実装と，Watts–Strogatz モデルをベースとした派生モデ

ルへの手法の適用などが考えられる．

謝辞 原稿を注意深くお読み頂き，適切な助言を頂きました査読者および編集委員に感謝致します．本研究は JSPS 科研費 JP19K11901 の助成を受けたものです．

参 考 文 献

[1] Barabáshi, A.-L. and Albert, R.: Emergence of Scaling in Random Networks, *Science*, No. 286 (1999), pp. 509–512.
[2] Barrat, A. and Weigt, M.: On the properties of small-world network models, *European Physical Journal B*, Vol. 13, No. 3 (2000), pp. 547–560.
[3] Bressan, S., Cuzzocrea, A., Karras, P., Lu, X., and Nobari, S. H.: An Effective and Efficient Parallel Approach for Random Graph Generation over GPUs, *Journal of Parallel and Distributed Computing*, Vol. 73, No. 3 (2013), pp. 303–316.
[4] Dean, J. and Ghemawat, S.: MapReduce: Simplified data processing on large clusters, *Communications of the ACM*, Vol. 51, No. 1 (2008), pp. 306–318.
[5] Erdös, P. and Rényi, A.: On random graphs I, *Publicationes Mathmaticae Debrecen*, Vol. 6 (1959), pp. 290–297.
[6] Klemm, K. and Eguíluz, V. M.: Highly clustered scale-free networks, *Physical Review E*, Vol. 65 (2002), pp. 036123.
[7] Leskovec, J., Chakrabarti, D., Kleinberg, J. M., Faloutsos, C., and Ghahramani, Z.: Kronecker Graphs: An Approach to Modeling Networks, *The Journal of Machine Learning Research*, Vol. 11 (2010), pp. 985–1042.
[8] 縄田和満: 基礎系数学確率・統計 *I* (東京大学工学教程), 丸善出版, 2013.
[9] NetworkX Developers: Overview of NetworkX. https://networkx.github.io/documentation/stable/.
[10] The Apache Software Foundation: Apache Hadoop. https://hadoop.apache.org/.
[11] Watts, D. J. and Strogatz, S. H.: Collective dynamics of 'small-world' networks, *Nature*, No. 393 (1998), pp. 440–442.

神野 薫

1996 年生．2019 年九州工業大学情報工学部知能情報工学科卒業．現在，同大学大学院情報工学府博士前期課程在籍．複雑ネットワークの生成，解析などに興味を持つ．

江本健斗

九州工業大学准教授 (2015 年 4 月より)，博士 (情報理工学)．高水準並列プログラミング手法，アルゴリズム導出，それらの定理証明支援系による形式的証明などに興味を持つ．日本ソフトウェア科学会，情報処理学会，ACM 各会員．