

シヨートノート

ランダムグラフの閉路数に関する一考察†

須崎 健 一††

本論はランダムグラフと閉路数の関係について考察する。ランダムグラフの性質は複雑で不明な点が多い。節と枝の接続関係が与えられた連結グラフの閉路数検出には、接続行列で求められる。しかし、ランダムグラフは生成されるまで接続関係が定まらず、ランダム枝集合より直接に閉路数の検出はできず、何らかの前処理が必要となる。したがって、本論では前処理の一手順として、ランダム枝の編集を行い、その枝編集より、すべての閉路数を検出した。また、本論はランダム枝の発生比率と閉路数の関係をシミュレーションし、発生比率が1.2%以上では、閉路数は指数関数的に増えてくることを示した。

1. ま え が き

近年、多くの組合せ論的な問題（例えば、巡回セールスマン問題、彩色数問題など NP 完全）に対して、近似解を求める多項式時間の算法とか、最適解を与える確率が高い多項式時間の算法などいくつか提案されている^{1),2)}。これら算法の有効性の評価は、数多くの計算実験に頼らざるをえなく難しい問題である。このようにときにランダムなデータで構成されるランダムグラフ⁵⁾⁻⁸⁾ (Random Graph: 以下、単に RG と略す)が必要となる。また、算法の複雑度は実用上、最悪値での評価よりも平均的な評価が望ましい場合が多い^{3),4)}。これら算法の複雑度の平均的な評価を行う上でも RG の性質を明らかにすることが必要である。

Erdős, Rényi ら^{5),6)}は、節点数が十分大きい条件のもとで RG が連結になる確率を導き、これらの性質から RG の連結成分の大部分が孤立点になることを明らかにした。また、高見沢ら⁷⁾は、大部分の節点の一つの非可成分に含まれており、残りの非可成分は、ほとんど1本の枝だけからなる成分であることを明らかにした。RG の性質は複雑でかつ不明な点が多く、その一つとして RG の中に含まれる閉路数に関する統計処理などがある。従来、連結グラフの閉路数検出には接続行列より探し出す算法がある。しかし、この算法は節と枝の接続関係が与えられた連結グラフには有効であるが、RG のような生成の度ごとにデータ構造が変化するような自由度の大きいグラフに

は有効ではない。また、閉路数の統計処理には数多くの RG を扱う理由から、生成ごとに接続行列を作成するよりは枝の編集から閉路数の検出を行う方法が処理手順の面から有効である。枝の編集とは、ランダム枝集合の任意の節より枝を並べ替えることをいう。閉路数は枝編集を用いて、閉路を含む部分グラフに相当する枝の並びを抽出し、バックトラック法¹⁰⁾で検出している。さらに RG 全体の閉路数を求めている。著者はランダム枝の発生、枝の編集、RG の生成及び、閉路数の検出等を行う処理プログラムを作成した。本論はこれらのプログラムを用いて、ランダム枝の発生比率と閉路数の関係を示した。

2. ランダム枝編集

RG の定義は文献5)に従い、本論で扱うランダム枝集合とは、ラベルの付けられた m 個の節のいずれか2点を結ぶ総数 mC_2 本の枝の中から、ランダムに抽出した n 本の枝集合とする。また、ランダム枝を (l, r) で表し、かっこの第1項目の l を左節名、第2項目の r を右節名と呼ぶ。ランダム枝の編集とは、ランダム枝集合の任意の節を指定したのち、その節から出る枝を、木のなぞり方で知られているプレオーダー⁹⁾の順に枝を並べ替えることをいう。ここで、プレオーダーのなぞり方は、節に立ち寄り、左の部分木をなぞる、次に右の部分木をなぞるの順で、木のすべての節をなぞる算法であるが、閉路を含むような RG ではなぞりがループとなり、なぞりが完遂しない。したがって、本論では一度なぞった枝には処理済のマークを付け、再び同じ枝をなぞらないようにしている。例えば、図5の RG (この RG はランダム枝の編集の結果、生成されるのであって、ランダム枝集合より直接

† Consideration on the Number of the Closed Loops of a Random Graph by KENICHI SUZAKI (Fukuoka Institute of Technology).

†† 福岡工業大学通信工学科

lft	8	5	1	0	6	2	2	1	7	7	2	1	3	6	1	8	7	7	6	9	0	9	2	1	9	6	7	2	1	4	7	9	6	
rgt	3	6	5	8	0	5	3	6	6	2	1	0	1	5	3	8	1	0	8	1	8	5	9	6	8	5	0	5	0	9	0	5	0	9

図 1 ランダム枝集合

Fig. 1 Branch set of randomness.

lft,rgt:ランダム枝集合の左・右節名配列 (図1)
 wl,wr :前処理後の枝の格納配列 (図3)
 k :左節名が共有している枝の有無フラグ

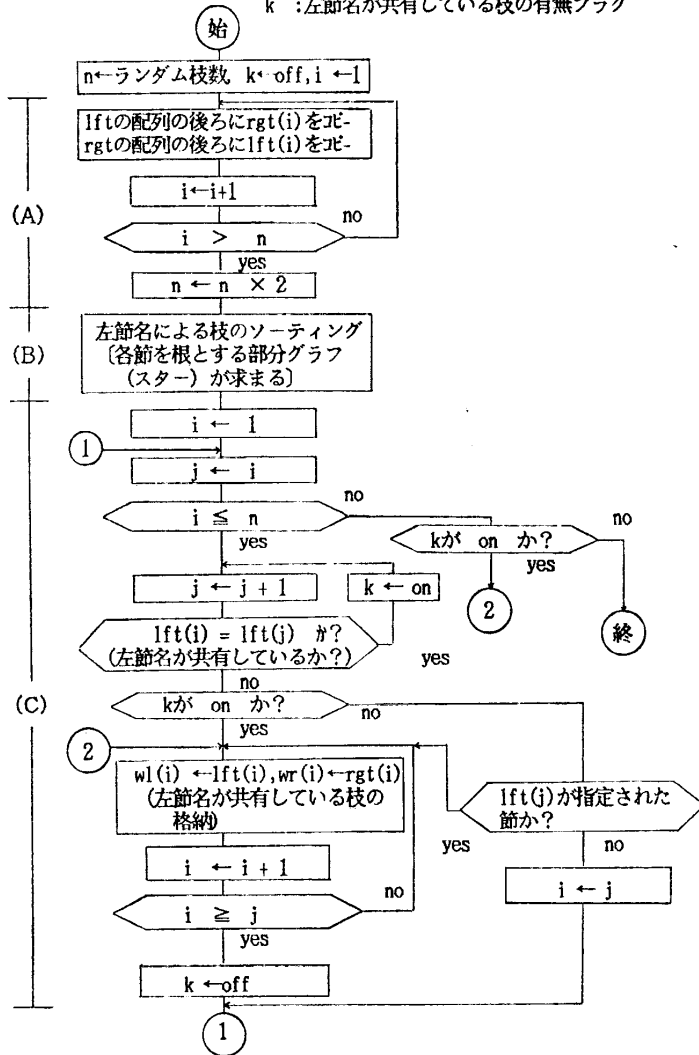


図 2 枝編集の前処理のアルゴリズム

Fig. 2 Algorithm for pre-processing of branch editing.

wl	0	5	0	5	0	9	0	9	0	9	1	0	1	0	1	0	1	8	1	8	2	1	2	1	2	1	2	1	3	6	3	6		
wr	6	2	7	7	4	7	8	5	2	1	9	6	5	8	2	1	7	7	8	6	8	8	1	0	9	6	0	9	3	6	8	5	2	1

wl	3	6	3	8	4	7	5	8	6	2	6	2	6	9	7	7	7	7	7	8	1	8	5	8	5	8	5	9	6	9	6	9	6	
wr	1	5	1	8	0	5	1	0	5	7	7	8	1	6	2	1	0	0	5	6	8	9	0	9	3	6	9	6	8	5	0	9	2	1

図 3 逆枝追加, 左節名による枝のソーティング, 左節名が共有していない枝の削除

Fig. 3 Addition to inverse branches, sorting by left node name, deletion of multiple branches, and the branch set from which the branches without a multiple left node name are deleted.

に生成されたものではない) をなぞると, まず指定された節 '10' から, (10,58) (10,21) (21,96) ... (05,47) となぞり, その結果, 図 4 の枝編集が得られる. 枝編集の際, 指定した節を変更すると, なぞり方が変わり図 4 とは別のものが枝編集される. しかし, これは閉路の検出に何ら影響を与えない.

さて, 図 1 のようなランダム枝を発生させ, 図 4 のごとく枝編集を行うわけであるが, ランダム枝のため, 直接に枝編集すると一枝一枝のなぞりに対して, すべての枝を調べていくことになる. したがって, ランダム枝に対して, 何らかの前処理が必要となる. ここでは, 任意の節を指定後, 図 2 に示すごとく前処理を行う. 図 2 の (A) 部分は, n 本のランダム枝 (l, r) に対して, 逆方向の枝 (r, l) を生成・追加する. (B) 部分は, 左節名による枝のソーティングを行い, 各節を根とみなす距離 1 の部分グラフに相当する枝並びを求める. その結果, 枝並びの中には, 左節名がいくつか共有する枝, または, 共有していない枝に分けられる. 例えば図 5 の RG において, 節 '58', '15' 及び '47' を左節名とする枝, つまり (58, 10), (15, 36) 及び (47, 05) の枝は左節名が 1 個で, 左節名を共有していない枝と呼ぶ. 一方, その他の節を左節名とする枝は 2 個以上の左節名が存在することになり (節 '77' だと, (77, 10), (77, 62), (77, 05) の 3 本), 左節名を共有する枝と呼ぶ. 次の (C) 部分では, 部分グラフに相当する枝並びにおいて, 左節名が共有していない枝の削除を行う. このとき, 削除される枝には, 上記で述べた枝以外に, RG に非連結な枝, 例えば図 1 のランダム枝の中には, 図 5 の RG に非連結な枝 (18, 38) と (69, 81) も削除される. さらに, (A) 部分で逆方向の枝を生成しているので (38, 18) と (81, 69) の枝も同時に削除され, 図 3 のごとくなる. 図 3 は左節名が共有している枝並びを配列 wl, wr に格納している. ここで, %印は削除した枝を表す. 図 4 は図 3 より前述したごとく, 枝をなぞりながら枝編集した結果である. 図 5 は図 4 より簡単なポインタ操作で生成した RG である. ここで, 枝編集までに要する手数は, ランダ

pl	1	0	1	0	2	1	9	6	8	5	0	9	2	1	3	6	3	6	0	9	1	0	7	7	6	2	0	5	0	5
pr	5	8	2	1	9	6	8	5	0	9	2	1	3	6	8	5	1	5	9	6	7	7	6	2	0	5	7	7	4	7

図4 枝編集
Fig. 4 Branch editing.

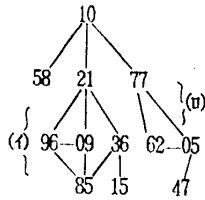


図5 枝編集より生成したランダムグラフ
Fig. 5 Random graph generated through editing.

ム枝数 n を発生するのに、 $O(n^2)$ 、前処理に $O(n \log n)$ 、枝編集に $O(n \log n)$ のオーダになる。

3. 閉路数の検出

閉路数の検出手順は、まず図4の枝編集より左節名と同じ節名をもつ右節名を探索し、図6に示すごとく、閉路を含むいくつかの枝グループをまとめて抽出する。図6の(イ)の例では3個の閉路が含まれている。しかし、実際には7個の閉路が存在する。したがって、これらを探し出すには、(イ)の枝グループに対して、枝編集を行った際に用いた前処理を行う。その結果、図7に示すごとく閉路グループ内のすべての節からの部分グラフ(スター)が得られる。この部分グラフより、バックトラック法で、閉路グループ内のすべての閉路数を検出する。

図8は閉路グループより、すべての閉路数を検出する流れ図である。閉路の判定は、各枝に優先順位などないため、閉路の抽出ごとに枝を円順列に並べ替えて

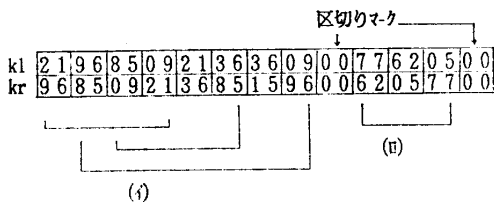


図6 閉路グループの枝抽出
Fig. 6 Branch set extracting the group of branches including closed loops.

sl	0	9	0	9	0	9	2	1	2	1	2	1	3	6	3	6	3	6	8	5	8	5	8	5	9	6	9	6	9	6
sr	2	1	9	6	8	5	9	6	3	6	0	9	8	5	1	5	2	1	0	9	9	6	3	6	8	5	2	1	0	9

図7 閉路グループ配列(閉路検出の前処理)
Fig. 7 Pre-processing to find out closed loops.

から、検出済の閉路(閉路収容領域内の閉路)と比較し、一致するものがないなら、新しい閉路と判定し閉路収容領域に格納する。

以上の処理を閉路グループごとに行い、RG全体の閉路数を求めた。本論では実際にシミュレーションし正確に閉路数が検出されたことを確認している。

4. シミュレーション

図9はランダム枝数 (n) の発生比率 (X) と閉路数 (Z) の関係を (Y) 回求め、3次元表示したものである。ランダム枝数 (n) 本の発生は、発生の度ごと(図9では (X) が 1.1~1.68% までの30回の発生、この関係を50回 (Y) で、全部で1,500回の発生)に、一様乱数の初期値を変更(一様乱数で変更)し、同じパターンのランダム枝が発生しないようにした。 (X) は $(n/mC_2) \cdot 100$ で表し、節点数 m は、100個とした。 (Z) は生成されたRGの個数に対して、各RGごと

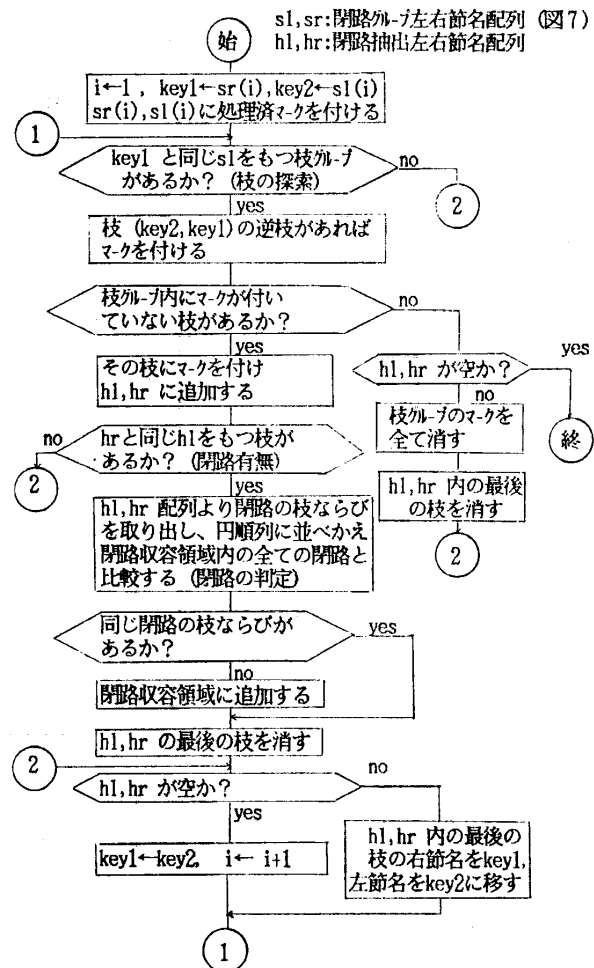


図8 閉路検出処理のアルゴリズム
Fig. 8 Algorithm for closed loop detection procedure.

- 6) Erdős, P. and Rényi, A.: On the Existence of a Factor of Degree One of a Connected Random Graph, *Acta Math. Acad. Sci. Hung.*, Vol. 17, pp. 359-368 (1966).
- 7) 高見沢, 滝内, 西関, 齊藤: ランダムグラフの統計解析, *情報処理*, Vol. 19, No. 8, pp. 742-748 (1978).
- 8) 五十嵐: 確率的アルゴリズムの概観, *情報処理*, Vol. 21, No. 1, pp. 13-18 (1980).
- 9) Knuth, D. E.: *The Art of Computer Programming*, Vol. 1, *Fundamental Algorithms*, Addison-Wesley, pp. 315-332 (1968).
- 10) 大附: アルゴリズムの複雑度の理論, *電子通信学会誌*, Vol. 62, No. 7, pp. 789-798 (1979).

(昭和60年8月8日受付)

(昭和61年2月20日採録)



須崎 健一 (正会員)

昭和23年生. 昭和47年日本大学理工学部電気工学科卒業. 昭和52年東京電機大学大学院修士課程修了. 昭和47年防衛大学校電気工学教室助手. 昭和55年福岡工業大学通信工学科講師となり現在に至る. グラフ理論, 分散型人工知能の分野に興味をもつ. 電子通信学会会員.

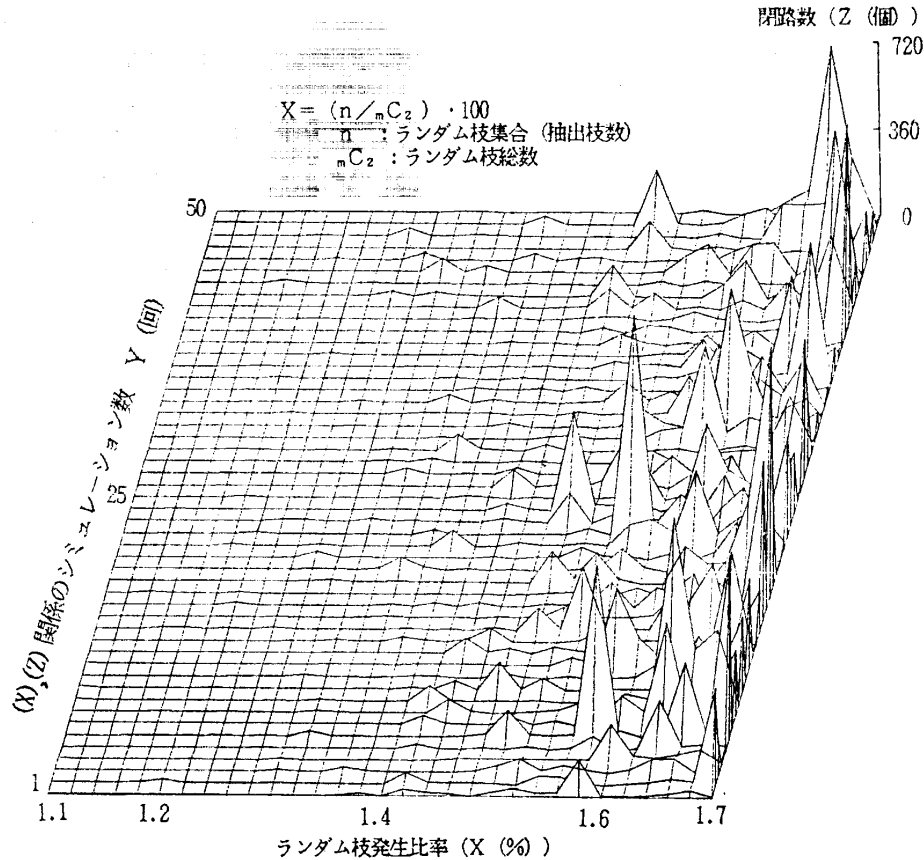


図 9 ランダム枝集合の発生回数, ランダム枝発生比率および閉路数の関係
 Fig. 9 Relationships among the frequency of random branches simulation (Y), ratio of random branch occurrence (X), and number of closed loops (Z).

に閉路数を求め, その合計とした. 図9より, (X) が 1.2, 1.4, 1.6 及び 1.68% と増えるにしたがい, (Z) の平均値は 1, 15, 80 及び 140 個と増えている. また, 図では現れていないがシミュレーションでは, (X) が 1.8, 2, 2.2% で, (Z) の平均値は 380, 1010, 2910 個とほぼ指数関数的に増加する傾向を示した.

5. むすび

ランダム枝集合より生成される RG の閉路数について考察した. 閉路数の検出は, ランダム枝編集より閉路グループの枝抽出を行い, グループごとに閉路を求め全体の閉路数を求めていく手順で行った. この処理手順は RG の一部が変化しても, 部分的な枝編集を再編集するだけでよく, また, 節と枝の接続関係が与えられた連結グラフに対しても, 各枝をランダム枝とみなせば, 閉路数の検出ができる. 本論で示した RG と閉路数の関係は, ランダム枝の発生比率が増えるとともに, 閉路数が指数関数的に増大していく傾向があることを示した. さらに, 詳しい関係を導くため

には, 節点数と発生比率の増加によるシミュレーション及び閉路数の発生確率などの統計処理が必要となる.

参 考 文 献

- 1) Garey, M. R. and Johnson, D.S.: The Complexity of Near-optimal Graph Coloring, *J. ACM*, Vol. 23, pp. 43-49 (1976).
- 2) Garey, M.R., Graham, R.L. and Johnson, D.S.: Some NP-complete Geometric Problems, *Proc. 8th ACM Symposium on Theory of Computing*, pp. 10-22 (1976).
- 3) Graham, R.L.: Bounds for Certain Multiprocessing Anomalies, *Bell Syst. Tech. J.*, Vol. 45, pp. 1563-1581 (1966).
- 4) Angluin, D. and Valiant, L.G.: Fast Probabilistic Algorithms for Hamiltonian Circuits and Matchings, *Proc. 9th ACM Symposium on Theory of Computing*, pp. 30-41 (1977).
- 5) Erdős, P. and Rényi, A.: On Random Graphs I, *Publicationes Mathematicae*, Vol. 6, pp. 290-297 (1959).