解説

グラフクラスとアルゴリズム

Graph Classes and Algorithms

上原隆平

Abstract
計算機で扱う問題は，多くの場合グラフ上の問題として定式化できる．計算量の理論により，これまで多くの問題が“手に負えない”ことが示されてきた．一方でこうした問題に対する現実的なアプローチが幾つか提案されてきた．本稿ではグラフに制限を加えるアプローチについて解説する．DNA の切片間の関係などは，モデル化すると特別なグラフになる．こうしたグラフ上では，これまで手に負えないとされてきた問題が効率良く解けることがある．本稿では，代表的なグラフクラスと，関連したアルゴリズムの最近の研究動向を解説する．
キーワード：アルゴリズム，グラフクラス，計算の複雑さ，理想グラフ

1. はじめに

本稿では，様々なグラフのクラスと，これに関連したアルゴリズムの研究動向を解説する．“グラフのアルゴリズム”といっても，もちろん「表計算ソフトで棒グラフや円グラフを描く方法」といった話ではない．世の中の森羅万象を頂点と呼ばれる“点”と辺と呼ばれる“線”を使ってモデル化する，数学でいうところのグラフである．「グラフ理論なんて数学じゃない」という数学者も，最近はめっきり少なくなってきたので，本稿で「数学でいうところのグラフ」といっても差し支えなかろう．

さて，世の中の多くの問題は，グラフ上の問題として定式化できる．これまで，計算量の理論と呼ばれる分野では，こうしたグラフ上の問題の多くが，どんなに高速なコンピュータを使っても“手に負えない”問題であることを示してきた．典型的な問題としては次の「ハミルトン路問題」が挙げられる．

入力：都市の集合と，都市間をつなぐ道の集合
出力：すべての都市を丁度 1 度ずつ通る経路があれば，それを見つけて出力

ここでは各都市が頂点であり，二つの都市を結ぶ道が辺に該当する．原理的には，候補となる経路をすべてチェックすれば答えは必ず見つけることができる．数学的には面白くも何ともない問題である．またコンピュータで単純にチェックを行うプログラムを作るのも簡単である．しかし，ちょっと都市の数が大きくなると，こうしたプログラムでは現実的な時間で答えを見つけることは不可能になる．この問題は計算量の理論では NP 完全問題と呼ばれており，どんなに工夫して高速化しても，都市の数が大きくなると，一般には実際的な時間で答えを見つけることはできない．これが計算量の理論における結論である．

理論的にダメだ，といわれて，そこであきらめることができるならよいのだが，現実問題としてはそうはいかない．そこで，様々な「現実的」な枠組みが提案され，理論的に研究されてきた．特に最近では，

- 近似アルゴリズム：(特に近似率の上界が得られれば) 近似解でも許す[1]
- 乱択アルゴリズム：乱数を使って確率的なアルゴリズムで計算する[2]
- 問題の入力に制限を加える：問題に制限を加えて，その制限の下で効率良く解を求める

といった枠組みが活発に研究されている（もう一つ固定パラメータ容易性[3]という無視できない枠組みがあるのだが，ここではあえて無視する）．

本稿で紹介するのは，三つ目の枠組みである．つまり入力されるグラフに妥当な制限を加える，というアプローチである．例えばハミルトン路問題であれば「道の総数は都市の総数に比例する」とか「ある都市につながっている道の数はせいぜい 100 本」といった制限は，ある

上原隆平　正員　北陸先端科学技術大学院大学情報科学研究科
E-mail　uehara@jaist.ac.jp
Ryuhei UEHARA, Member (School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1291, Japan).
電子情報通信学会誌　Vol.88 No.2 pp.118-122 2005 年 2 月

程度の妥当性を持つ，自然な制限といえるであろう．このように入力に制限をつけたときに，それを使うことによって，“手に負えない”問題が“手に負える”問題になることがある

では，どんなグラフが「妥当性を持っている」とされるのか，またどんな問題がそのグラフ上で解けるのか．本稿ではそのごくごく一部を概観したい．

2. 区間表現を持つグラフ

本稿で扱っているグラフクラスとアルゴリズムに関する研究は，歴史的には1950年代後半に始まったといえる．当時，数学系の研究者であるHajösと，分子生物学者であるBenzerとが“区間グラフ”と呼ばれるグラフクラスを独立に考えたことに端を発している(文献(4)，Chapter 8, pp.171-172参照)．区間グラフでは，各頂点は「数直線上のある区間」に対応する．そして二つの区間が重なっているとき，対応する二つの頂点は辺で結ばれる．

ここでまず区間グラフと分子生物学の関連について補足しておこう．全ゲノムショットガン法と呼ばれる解析方法では，まずゲノムをたくさんの断片に分ける．そしてこれらの断片同士が「重なり」を持つかどうかという情報を使って元のゲノムの配列を構築する．この場合，ゲノムの断片を文字列上の区間ととらえれば，断片の情報を表すグラフは区間グラフとなる(図1)．例えば四つの断片 x_1，x_2，x_3，x_4 の重なりを調べた結果，x_2 と x_4 のみ重なりを持たないことが分かったとする．このときこの四つの断片を頂点とすると，対応する区間グラフ(b)が得られる．この区間グラフ(b)は(c)に示した区間表現を持つ．

区間グラフは実生活の中でもよく見られるグラフである．例えば工程表の例を考えてみよう(図2)．各仕事ごとに作業時間を割り当て(例えば仕事aは8:00～11:00，仕事hは16:00～19:00など)，これを区間，

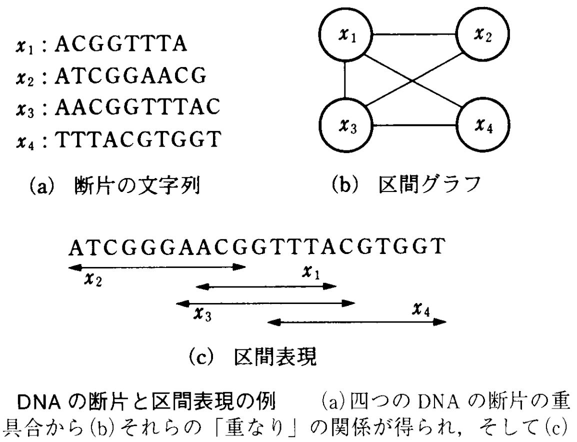

図1 DNAの断片と区間表現の例 (a)四つのDNAの断片の重なり具合から(b)それらの「重なり」の関係が得られ，そして(c)元のDNAの文字列が得られる．

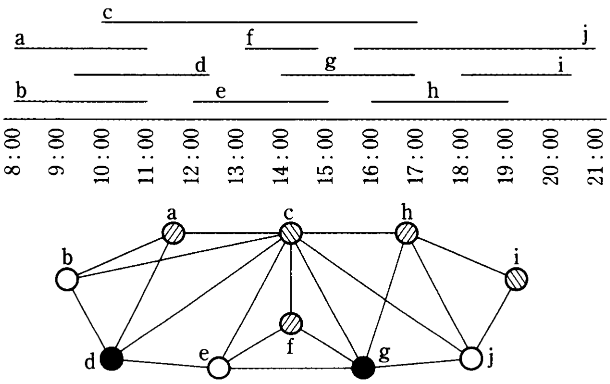

図2 工程表の例 仕事aは8:00～11:00まで，といった具合．時間が重なる仕事は同時に一人の人間が行うことはできない．では何人必要なのか？あるいは，何人いれば十分なのか？

すなわちグラフの頂点と見なすのである．この場合，区間グラフの頂点はそれぞれの仕事に対応し，例えば頂点aと頂点bの間に辺があるときは(これらは時間が重なるので)，一人の人間は仕事aと仕事bを担当することができない，ということになる．

こうした工程表を実行するに当たり，作業員は最低何人いれば足りるのであろうか．ある頂点集合 C を見たとき，どの二つの頂点間も辺で結ばれているなら，C をクリーク(clique)と呼ぶ．他のクリークに含まれていないクリークを極大クリークといい，頂点数が最も多いクリークを最大クリークという．このとき，すぐに次のことが分かる：『グラフがクリーク C を含むなら，最低でも $|C|$ 人の作業員が必要である．』($|C|$ はクリーク C に含まれる頂点の数を表す．)

つまり，工程表に対応する区間グラフが与えられたとき，最大クリークを求めれば，少なくともそれだけの人数は必要であることは分かる．実は区間グラフではこうした最大クリークの大きさは簡単に求めることができる．図2の工程表であれば，各時刻に『幾つの仕事が同時に行われているのか』を順番に見ていけばよい．すると {a,b,c,d} や {c,g,j,h} など，大きさ4のクリークがあることが分かる．

ではこの場合，4人いれば十分なのであろうか．もし4人で十分であれば，元の区間グラフを『どの辺の両側も色が違う』ように頂点を4色で塗り分けることができるはずである．このように『どの辺の両側も色が違う』ような塗り分けに必要な，最小の色の数を彩色数と呼ぶ．実は区間グラフでは，最大クリークの大きさと彩色数が一致することが分かっている．図2の工程表では例えば {a,f,h}，{b,e,j}，{c,i}，{d,g} と塗り分ければよい．つまり，これらの仕事の集合にそれぞれ担当する作業員を割り当てることにすれば，図2の工程をこなすには4人いれば十分であることが分かる．

なお，一般のグラフでは「最大クリークの大きさ」よりも「彩色数」の方が大きくなる．図3のグラフは，最

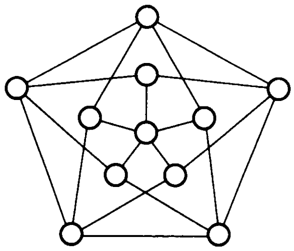

図3 最大クリークの大きさは2で，彩色数は4のグラフ もしこれが工程表なら，一見二人で仕事がこなせそうなのに，実際には四人いないと仕事がこなせない．

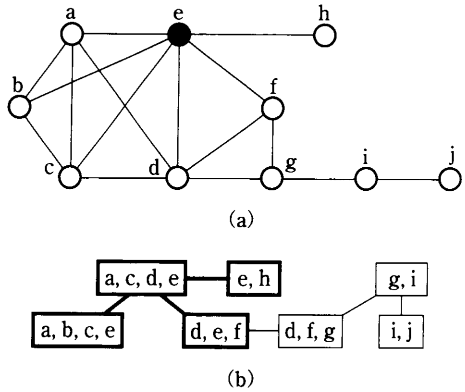

図4 弦グラフの例 (a)で与えられる弦グラフのクリーク木の一例が(b)．クリーク木は一意的に決まるわけではない．

大クリークの大きさは2であるが，塗り分けには4色必要である．つまり図3のグラフは区間グラフではない．

このように区間グラフは日常様々な場所で現れるが，有用な性質を多数持っているため，一般には手に負えない様々な問題を簡単に解くことができる．例えばハミルトン路問題も簡単に解くことができる[(5)]．昨今では特にDNA 関連の問題が，区間グラフモデルや，更にそれを一般化したグラフモデル[(6)]の上で考えられ，研究されている．

3. 木表現を持つグラフ

頂点が循環的につながっている場合，例えば頂点 v_1，v_2，v_3，v_4，v_5 がそれぞれ辺 $\{v_1, v_2\}$，$\{v_2, v_3\}$，$\{v_3, v_4\}$，$\{v_4, v_5\}$，$\{v_5, v_1\}$ でつながっている場合，これをサイクルと呼ぶ．区間グラフ上で長さ4のサイクル $(v_1, v_2, v_3, v_4, v_1)$ を考えると，どんな区間表現を考えても，$\{v_1, v_3\}$ か，あるいは $\{v_2, v_4\}$ が辺となる．こうした「サイクル上にあり，かつサイクルを構成しない辺」のことを弦と呼ぶ．弦グラフ（chordal graph）とは，「長さ4以上のサイクルは必ず弦を持つ」という特徴を持つグラフである(図4 (a))．

弦グラフは，1970年代に様々な分野で独立に考えられたグラフクラスである．そのため，弦グラフ（chordal graph）以外にも，triangulated graph，rigid circuit graph などと呼ばれていて，どの呼び名を使うかで，その人の出身や年代が分かる（文献(4)，Epilogue，p.281 参照)．

弦グラフはこうしたサイクルを用いた定義以外にも，幾つかの同等な定義があることが知られている．特に美しいのは，次の定義である．まず木 T を考える．なお，木とはサイクルを含まない連結なグラフのことである．各頂点 v はそれぞれ，T の部分木 T_v，つまり T の一部の頂点からなる小さな木 T_v に対応する．そして頂点 u と頂点 v は，対応する部分木 T_u と T_v が重なりを持つときに辺で結ばれる．（これは区間グラフの一般化にもなっている．つまり区間グラフは弦グラフである．）例えば図4 (a)の弦グラフに対応する木 T の一例が図4 (b)である．木の頂点 N の中に弦グラフの頂点 v が書かれているとき，頂点 N は部分木 T_v に含まれていることを意味している．例として元の弦グラフの頂点 e に対応する部分木 T_e を太線で示した．

この定義は単に美しいだけではなく，アルゴリズムを構成する上で有用な多くの性質を持つ．特に木 T のそれぞれの頂点は，元の弦グラフの極大クリークと1対1で対応付けることができる．（正確にいうと，そうなるように T を作ることができる．）こうした性質を持つ木 T をクリーク木と呼ぶ．図4 (b)の木は，クリーク木の一つである．このクリーク木から，図4 (a)の弦グラフの極大クリークは七つあり，最大クリークは大きさ4のものが二つあることが分かる．そしてそれらは具体的に{a,b,c,e} と {a,c,d,e} の二つであることも分かる．

一方，弦グラフは頂点の順序付けで定義されることもある．ある頂点 v と，その隣接点の集合 $N(v)$ が，グラフ上でクリークになっているとき，その頂点を simplicial である，という．また与えられたグラフ G に対して，一部の頂点を取り出し，G の辺をそのまま引き継いで構成したグラフを導出部分グラフと呼ぶ．頂点の並び v_1，v_2，…，v_n が「各頂点 v_i は $\{v_i, v_{i+1}, \cdots, v_n\}$ に導出されるグラフ上で simplicial である」という条件を満たすとき，そのグラフは弦グラフであり，その逆もまた成り立つ．つまりこの頂点の並びによって，弦グラフを定義することができる．この並びのことを perfect elimination ordering と呼ぶ．例えば図4 (a)の弦グラフであれば，j，i，h，g，f，d，e，c，b，a は perfect elimination ordering である．

与えられたグラフが弦グラフであるかどうかを判定するアルゴリズムは，1970年代に幾つか提案された．特に Rose，Tarjan，Lueker によって文献(7)で提案されたアルゴリズムは LexBFS（Lexicographically Breadth First Search：辞書式順序に基づいた幅優先探索）と呼ばれ，名前は複雑だが，アルゴリズム自体は非常に単純で，かつ高速に動作する．実装も本当にやさしい．LexBFS は与えられたグラフが弦グラフのときは，perfect elimination orderingを出力する．1990年代には，このperfect elimination orderingからクリーク木を構築する，単純かつ高速なアルゴリズムも幾つか示されている[8]．（なおLexBFSそのものも，いろいろなグラフクラスの認識に使える，ということで最近一部で熱心に研究されている[9]．）

これらの多彩な特徴付けを使うと，弦グラフ上ではいろいろな問題を効率良く解くことができる．例えば最大クリーク，最大の独立点集合（互いに辺で結ばれていない頂点集合），彩色数などが簡単に求めることができる[10]．一方ではハミルトン路問題は弦グラフ上でもNP困難であり，依然として手に負えない問題であることも知られている[11]．

つまり同じ“手に負えない”問題であっても，どのグラフクラスで簡単になるのか，という境界線は，問題によって（かなり）違ってくるのである．この境界線を明確にする，ということは，とりも直さずその問題の「どこが本質的に難しいのか？」という点を明確にすることにもつながる．

4. 理想グラフ

既に見たように，一般のグラフでは，最大クリークの大きさは彩色数よりも小さい．しかし理想グラフ（perfect graph）と呼ばれているグラフでは，この二つが一致する．より正確にいえば，理想グラフとは『すべての導出部分グラフにおいて最大クリークの大きさと彩色数が一致するグラフ』のことである．定義がかなり分かりにくいが，区間グラフも弦グラフも理想グラフのごく一部を占めているにすぎない．理想グラフとは，これらを含むかなり大きなグラフのクラスである．

理想グラフ上では，彩色数や最大クリークの大きさ，最大独立点集合などを効率良く見つけることができる．これらの問題は，代表的なNP完全問題であり，“手に負えない”問題である．こうした問題が効率良く解けるという，好ましい性質と，グラフのクラスが集合として大きいという性質とから，理想グラフは幅広く研究されてきた．

理想グラフの定義は上で示したとおりなのであるが，実はもっと単純な特徴付けと分かりやすい定義があるのではないか，という予想が40年以上も前からいわれていた．1960年にBergeは『あるグラフGが理想グラフである』ということと，『Gと$\overline{G}$とが，長さが5以上の奇数長の「弦を持たないサイクル」を導出部分グラフとして持たない』ということが同値である，と予想した．（$\overline{G}$はGの辺の有無を反転したもので，補グラフと呼ばれる．）これは「強理想グラフ予想」と呼ばれていた．そして2002年5月に「強理想グラフ予想」は「強理想グラフ定理」と呼ばれるようになった．Chudnovsky，Seymour，Robertson，Thomasらが強理想グラフ予想の証明に成功したのである（詳細は文献(12)に詳しい）．

この結果により，今後は理想グラフの定義，あるいは性質として「強理想グラフ定理」が使えるようになった．この性質は，もともとの定義や特徴付けに比べるとずっと単純であり，直観的にも分かりやすい．そのため，アルゴリズムを構成するときの「入力グラフが満たす性質」としても使いやすいように思う．今後は，もっと多くの“手に負えない”問題が，理想グラフという大きなグラフクラス上で，実際的な時間で解けるようになることが期待できる．

5. 蛇　　　足

本稿で紹介したグラフのクラスは，どちらかといえばグラフ理論からのアプローチである．そのため，グラフの性質をいわば静的に定義してきた．これとはやや趣が異なるアプローチとして，グラフをある条件の下でランダムに生成する，というモデルもある．これは1960年代に提案されたRandom Graphという概念がその始まりであり，特に1980年代以降，活発に研究された分野である．通常Random Graphといったときは，各辺が確率pで独立に存在するというモデルを使うことが多い．このため，生成されるグラフは一様な構造となる[13]．

近年WWWに代表されるインターネット上のサービスは，私たちの日常に深く入り込んでいる．しかしインターネットのネットワーク構造は，Random Graphのような一様性を持っているようにはとても見えない．筆者の使っている貧弱な端末と，日本の基幹となるDNSサーバとでは，ネットワークへの接続の“太さ”はまったく違っている．

もう少し具体的にいうと，インターネットをモデル化するグラフでは，一部の少数の頂点には辺が集中し，多数の頂点にはそれほど辺が集中していない，という性質を持っていなければ，説得力がない．こうした『ごく一部に接続が集中する』という現象をうまく説明できるモデルとして，近年Power Lawと呼ばれる法則が注目を集めている．（これはべき法則，べき乗則などと訳されている．）こうしたPower Lawを満たすグラフを確率的に生成する方法も提案されている．

こうした“Web Graph”の理論的な研究は，Webの普及とともに1990年代後半に始まったばかりである．しかし目ざとい研究者たちが，最近ではかなり精力的に研究を行っている．（文献(14)などが参考になる．）この分野も非常にエキサイティングな分野ではないか，と筆者は注目している．

6. 文献紹介

グラフ理論については，和書，洋書ともに最近は優れた教科書が幾つもある．しかし，あるグラフのクラスが「なぜ導入されたのか？」「どういう問題に使えるのか？」という素朴な疑問に答えるものはそれほど多くはない．文献(15)は，それぞれのグラフクラスについて，こうした疑問に答えてくれる例が豊富に示されているので，初学者には読みやすい．また，計算量の理論についても優れた教科書が幾つもあるが，初学者にも読みやすい，という点で文献(16)を推す．この文献は最新の結果もさりげなく採り入れていて，例えば「固定パラメータ容易性」という概念を日本語でいち早く紹介しているのは，筆者が知る限りではこの本だけである．

本稿で扱った内容を詳しく紹介している文献としては文献(4)が挙げられる．1980年に刊行された同書の初版は，この分野の『教科書』として有名であった．今回の改定版では，その後の20年間における進展もかなり取り込まれているし，代表的な文献を紹介している点でも推薦できる．

文献(17)は，本稿で紹介しているグラフクラスの分野がどれほどの広がりを持つのかを把握するには，欠かせない．この本は「読む」本というよりは，関連文献を「調べる」ための本といえる．300ページほどの分量中に，紹介されているグラフクラスが160個余り，参照されている論文数が1,100本以上である．論文リストを軽く眺めるだけでも，1990年以降にこの分野がいかに活発に研究されてきたかがよく分かる．

文献(4)と文献(17)との中間的な本として文献(8)も挙げておく．最後の方を読むと「今どんな問題が研究されているのか？」という最前線の情報が得られる．

文献

(1) V.V. Vazirani, Approximation Algorithms, Springer, 2001.
(2) R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge, 1995.
(3) R.G. Downey and M.R. Fellows, Parameterized Complexity, Springer, 1999.
(4) M.C. Golumbic, "Algorithmic graph theory and perfect graphs," Annals of Discrete Mathematics 57, 2nd edition, Elsevier, 2004.
(5) P. Damaschke, "Paths in interval graphs and circular arc graphs," Discrete Math., vol.112, Issues 1-3, pp.49-64, 1993.
(6) R. Uehara, "Canonical data structure for interval probe graphs," ISAAC 2004, pp.859-870. Lecture Notes in Computer Science, vol.3341, Springer-Verlag, 2004.
(7) D.J. Rose, R.E. Tarjan, and G.S. Lueker, "Algorithmic aspects of vertex elimination on graphs," SIAM J. Comput., vol.5, no.2, pp.266-283, 1976.
(8) J.P. Spinrad, Efficient Graph Representations, American Mathematical Society, 2003.
(9) D.G. Corneil, "A simple 3-sweep LBFS algorithm for the recognition of unit interval graphs," Discrete Appl. Math., vol.138, no.3, pp.371-379, 2004.
(10) F. Gavril, "Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph," SIAM J. Comput., vol.1, no.2, pp.180-187, 1972.
(11) H. Müller, "Hamiltonian circuit in chordal bipartite graphs," Discrete Math., vol.156, Issues 1-3, pp.291-298, 1996.
(12) V. Chvátal, PERFECT PROBLEMS, 2003. http://www.cs.rutgers.edu/~chvatal/perfect/problems.html
(13) B. Bollobás, Random Graphs, 2nd edition, Cambridge University Press, 2001.
(14) A. Broder, R. Kumar, F. Maghoul, R. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure in the web," Comput. Netw., vol.33, Issues 1-5, pp.309-320, 2000. http://www9.org/w9cdrom/160/160.html
(15) 惠羅　博，土屋守正，グラフ理論，産業図書，1997.
(16) 岩間一雄，オートマトン・言語と計算理論，コロナ社，2003.
(17) A. Brandstädt, V.B. Le, and J.P. Spinrad, Graph Classes: A Survey, SIAM, 1999.

うえはら　りゅうへい
上原　隆平（正員）
平3電通大・博士前期課程情報工学専攻了．同年キヤノン(株)情報システム研究所．平5東京女子大・情報処理センター・助手．平10駒澤大・自然科学教室・講師，平13同助教授．平16北陸先端大・情報科学研究科・助教授，現在に至る．ACM，IEEE，EATCS等各会員．博士(理学)．