進化の系統樹と系統ネットワークに関する組合せ論

早水　桃子　(早稲田大学)*

1　はじめに

地球上に存在する多様な生物たちが，単一の共通祖先からどのように別々の系統に分岐して現在に至ったかという（想像上の）進化史を単純明快に記述する手段として，「系統樹」と呼ばれる木グラフは Charles Darwin の時代から広く使われてきた（図 1）．一方，観測や計測などに基づく生物学的なデータから正しい進化史を解明する際には木構造では記述できない複雑な情報の取り扱いが必発するため，系統樹を一般化して分岐だけでなく合流も表せるようにした「系統ネットワーク」という拡張版モデルを使いたいというニーズも古くからある．ところが，今日でも生物の進化を調べる研究において中心的な役割を果たしているのは系統樹に関するデータ解析技術の方であり，系統ネットワークは生物間の距離（非類似度）を可視化する際のツールとして限定的な場面で利用されているに過ぎない．長年のニーズと大きな科学的意義があるにも関わらず系統ネットワークを活用したデータ解析技術が未だに発展途上なのは，系統樹とは異なり系統ネットワークが網目のように複雑なネットワークも含む厄介なグラフのクラスであり，この大道具の性質や使い道に関する数学や計算機科学や統計科学の様々な問題が解決されないまま山積みになっているからである．

このような背景から，進化の系統樹から系統ネットワークへのパラダイム・シフトは数理生物学者 Joel E. Cohen の 2004 年の有名なエッセイ [3] の中でも数学と生物学で相互にイノベーションを触発しうる研究テーマの一覧に挙げられており，この 20 年で純粋・応用を問わず様々な数学分野の研究者たちが系統学に関する研究領域に続々と参入してきた．その中でも系統ネットワークに関する組合せ論は多くの新規参入者を巻き込んで成長し，ソフトウェア開発につながる実用的な成果も生み出してきた若く勢いのある研究エリアである．組合せ論的系統学の分野全体を俯瞰した包括的な解説は成書 [12, 18, 19] に譲ることにして，本講演では，この研究分野の一端を紹介するべく，講演者による系統ネットワークの構造定理とその応用に関する論文 [10] の内容を中心に概説する．

2　系統樹と系統ネットワーク

本稿を通じて，$X = \{1, \ldots, n\}$ は空でない有限集合とする（X は現存する幾つかの生物種の集合と解釈され，しばしば**ラベル集合**と呼ばれる）．生物の進化を記述するという問

* 〒169-8555 東京都新宿区大久保 3-4-1 早稲田大学理工学術院 基幹理工学部応用数理学科
e-mail: hayamizu@waseda.jp
本研究は科学技術振興機構 (JST) 戦略的基礎研究推進事業さきがけ研究 (JPMJPR16EB, JPMJPR1929) の助成を受けたものである．

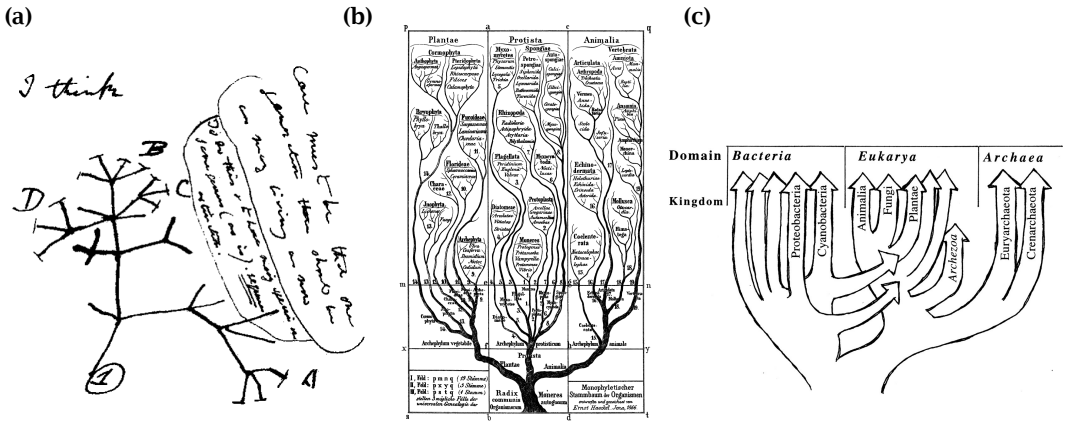

図 1　(a) 進化論の提唱者 Darwin が描いた “Tree of Life” の構想スケッチ（1837 年）．地球上の全生物の起源といえる共通祖先がかつて一つだけ存在し，そこから分岐を繰り返して現在の多様性が生まれたのだという学説．(b)Darwin を熱烈に支持して学説の普及を助けた生物学者 Haeckel が実際に作成した系統樹（1866 年）．(c)Science 誌に掲載された 1999 年の Doolittle の論文 [5] で「現在のスタンダード」として紹介されている進化のモデル．概形は系統樹に似ているが，頻繁ではないとはいえ分岐した道が再び合流している部分があるので系統ネットワークである．

題意識のため，本稿で考察する**グラフ／ネットワーク**（これら二つの用語は区別せずに用いる）は全て有限，単純，非巡回，有向グラフであり，各用語は次のように定義される．**有向グラフ**とは，頂点集合 V とアーク（有向辺）集合 A のペア (V,A) である．有向グラフ G の頂点集合は $V(G)$，アーク集合は $A(G)$ で表され，$V(G)$ と $A(G)$ がともに有限集合のとき，G は**有限グラフ** という．アーク a が頂点 u から頂点 v に向かうとき，a を (u,v) で表す．逆に，$tail(a)$ と $head(a)$ は，それぞれ u と v を表す．グラフ G が**単純グラフ**であるとは，任意の $a \in A(G)$ に対して $head(a) \neq tail(a)$ であり，かつ，任意の異なる $a, a' \in A(G)$ に対して $(head(a), tail(a)) \neq (head(a'), tail(a'))$ であることをいう．単純グラフ G が**非巡回**であるとは，G がサイクルを含まない，すなわち，任意の $i \in [1,k]$ に対して $head(a_{i-1}) = tail(a_i)$ であるような $A(G)$ の 3 つ以上の要素の列 $(a_0, a_1, \ldots, a_{k-1})$ が存在しないことをいう（ただし添字は $\bmod k$）．

グラフ G と H が与えられたとき，$V(G) \subseteq V(H)$ かつ $A(G) \subseteq A(H)$ ならば G は H の**部分グラフ**であるといい，$G \subseteq H$ と書く．部分グラフ $G \subseteq H$ は，H と同型でないときに H の**真部分グラフ**であるという．グラフ G とアーク集合 $A(G)$ の部分集合 A' が与えられたとき，A' は G の部分グラフ $G[A'] := (V(A'), A')$ を**誘導する**という．ただし，$V(A')$ は A' に含まれる全てのアークの head と tail の集合を表す．さらに，$|A(G)| \geq 1$ なるグラフ G と $A(G)$ の分割 $\{A_1, \ldots, A_\ell\}$ が与えられたときに，集合族 $\{G[A_1], \ldots, G[A_\ell]\}$ を G の**分解**という．なお，集合 S の**分割**とは，和集合が S になるような空でない互いに素な部分集合の集まりである．

有向グラフ N の頂点 v に対して，v の**入次数**と**出次数**は，それぞれ集合 $\{a \in A(N) \mid$

$head(a) = v\}$ と 集合 $\{a \in A(N) \mid head(a) = v\}$ の要素の数と定義され，それぞれ $deg^-_N(v)$ と $deg^+_N(v)$ で表される．非巡回グラフ N に対して，$(deg^-_N(v), deg^+_N(v)) = (1,0)$ を満たす頂点 $v \in V(N)$ は N の**リーフ**と呼び，N のリーフを全て集めたものを N の**リーフ集合**という．リーフという用語は木に対して用いられることが多いが，ここで述べたように系統学分野では木以外のグラフに対しても末端の頂点のことをリーフと呼ぶのが慣例となっている．

本稿ではこれまでに「系統ネットワーク」と「系統樹」という言葉を用いてきたが，以下ではこれらを数学的に定義する（系統ネットワークの具体例は図 3 の N を見よ）．

定義 2.1. ラベル集合 X に対し，次の条件 1–3 を満たす有限単純非巡回有向グラフ N を**（X 上の根付き二分）系統ネットワーク**と呼ぶ．

条件 1　N のリーフ集合とラベル集合 X の間に一対一の対応関係がある

条件 2　N は $deg^-_N(\rho) = 0$ かつ $deg^+_N(\rho) \in \{1,2\}$ を満たす頂点 ρ を唯一つ持つ

条件 3　N のリーフと ρ を除く任意の頂点 v は $\{deg^-_N(v), deg^+_N(v)\} = \{1,2\}$ を満たす

定義 2.1 の条件 2 における頂点 ρ は N の**ルート**といい，これは N のリーフに対応する現存種たちの共通祖先として解釈される．また，定義 2.1 の条件 3 における $(deg^-_N(v), deg^+_N(v)) = (2,1)$ の頂点を**合流点（*reticulation vertex*）**と呼ぶ [*1]．次の定義 2.2 を見ての通り，系統ネットワーク [*2] と系統樹の違いは合流点の有無だけである．

定義 2.2. ラベル集合 X に対し，定義 2.1 の条件 1 と条件 2 および，次の条件 3′ を満たす有限単純非巡回有向グラフ N を**（X 上の根付き二分）系統樹**と呼ぶ．

条件 3′　リーフでもルートでもない N の任意の頂点 v に対しては，$(deg^-_N(v), deg^+_N(v)) = (1,2)$ が成立する

3　Tree-based な系統ネットワークと全域系統樹

本稿での問題意識を説明するために，興味のある生物種の集合 X 上の根付き二分系統ネットワーク N が何らかのデータと何らかの方法を用いて構築されたが，N は合流点を持つため系統樹のように容易に解釈できず，X が辿った進化史について専門家でも意見が分かれてしまう状況を考えてみよう（図 2 参照）．この場合，N を眺めていても有意義な情報は得られそうにないので，N から余分なアークを削ぎ落として生物学的に解釈しやすい部分——N に含まれる X 上の根付き二分系統樹 T に相当する部分——を抽出するというのは現実的な方針である．そこで「N の余分なアーク」や「N に含まれる X 上の

[*1] 樹状ではなくネットワーク状の進化を意味する reticulate evolution の訳語として「網状進化」という用語が定着しているので，reticulation vertex も「網状点」と訳すのが自然かもしれないが，標準的な訳語が見当たらなかったため，ここでは理解しやすさを優先して「合流点」と呼ぶことにした（合流という解釈が妥当でない場合もあるので，本当は無理に和訳しないほうが安全である）

[*2] 本稿で取り扱う系統樹や系統ネットワークは全て「X 上の根付き二分」であるため，以後は混乱の恐れがなければ「系統樹」や「系統ネットワーク」という略称を用いても良いことにする．

根付き二分系統樹 T に相当する部分」を意味する用語を定義していこう．図 3 で例示した通り，本稿では，X 上の根付き二分系統ネットワーク N に X 上の何らかの根付き二分系統樹 $\widetilde{T}$ と位相同型な全域木 T が存在するときに「N には $\widetilde{T}$ が（T という形で）含まれる」と解釈する．そのような全域木 T を N の**全域系統樹** [*3] と呼び，全域系統樹を 1 つ以上持つ系統ネットワークを *tree-based network* と呼ぶ．

なお, 全域系統樹 T を題材とする諸問題を取り扱う本稿では T が主役となる tree-based network の定義を採用しているが，これと同値だが直観的に理解しやすいものとして，T と位相同型な $\widetilde{T}$ に着目した別の定義もある（図 4 参照）．

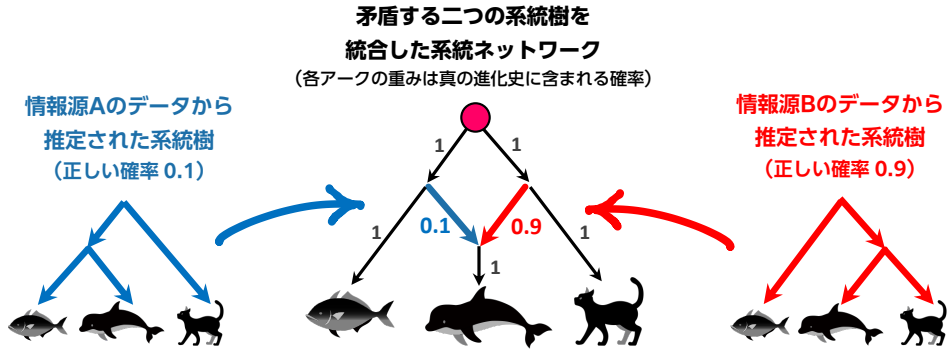

図 2　互いに矛盾する二つの系統樹を混合して作られた系統ネットワーク．この系統ネットワークはサイズが小さいので赤の系統樹（尤度 0.9）と青の系統樹（尤度 0.1）を復元することは容易にできるが，一般には混合前の系統樹を復元することは難しい．

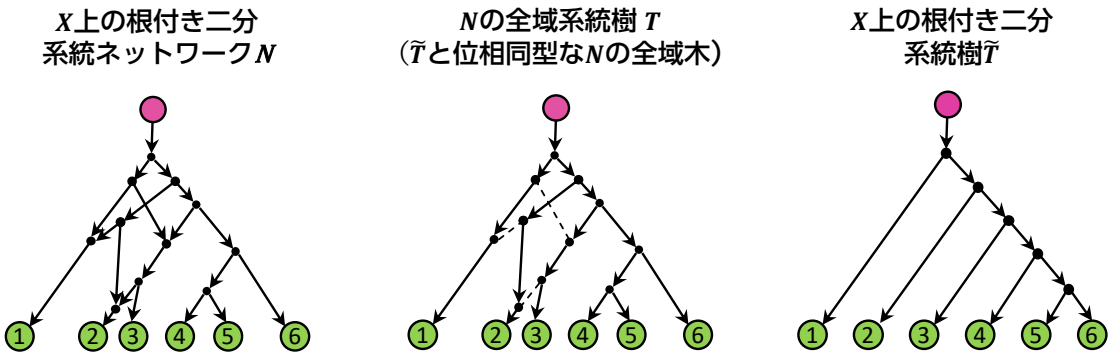

図 3　根付き二分系統ネットワーク N と全域系統樹 T の例（T のアークは実線部）．全域系統樹が少なくとも一つ存在するので N は tree-based network の例でもある．

Tree-based network は Francis と Steel の 2015 年の論文 [7] 以来，理論生物学のホットな研究トピックになっている（例：[7, 1, 21, 9, 2, 13, 16, 22]）．それは tree-based network がこれまでに組合せ論やアルゴリズムの視点で熱心に研究されてきた系統ネットワークの多様なサブクラスを含む一般的なものであることに加え，生物学者から見ても意味付

[*3] Tree-based network や全域系統樹は [7] で導入された概念である．ただし全域系統樹という日本語訳は筆者がつけたもので，英語では support tree や subdivision tree と呼ばれる．

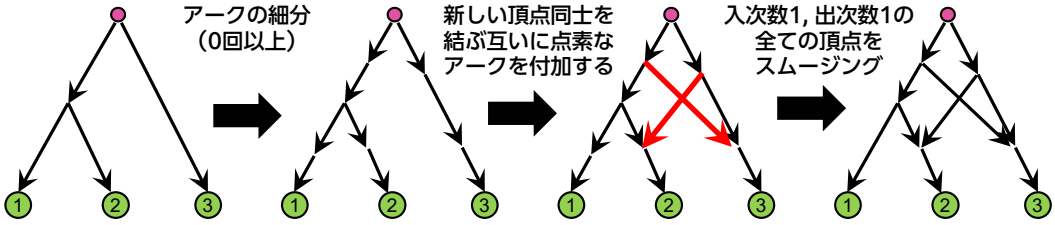

図 4　Tree-based network の構成方法に着目した別の同値な定義．図 3 の記法でいえば，一番左が $\tilde{T}$，一番右が N である．

けが容易で系統樹の自然な一般化と見なせる進化のモデルだからである．図 3 の通り N の全域系統樹 T が（したがってそれと位相同型な系統樹 $\tilde{T}$ が）存在するとき，N が $\tilde{T}$ に 0 本以上のアークを追加して作られる系統ネットワークなのは自明だが，N が「二分」系統ネットワークであることを思い出せば，$\tilde{T}$ に追加されたアーク同士が互いに端点を共有できないことにも気づく．つまり N の全域系統樹に含まれないアークというのは（あえて統計科学的な意味付けを試みるとすれば）真の系統樹 $\tilde{T}$ に付加された互いに独立なノイズのようなものであり，N の全域系統樹というのは N に含まれる独立ノイズを除去して真の系統樹 $\tilde{T}$ に相当する成分だけを抽出したものといえるだろう．

4　全域系統樹に関する基本的な計算問題

Tree-based network は系統ネットワークの proper なサブクラスであり，例えば図 5 の左に描かれた系統ネットワークは全域系統樹を持たない．すると，まずは以下の**決定／探索問題**を論じるのが順当であろう．

問題 1 ([7])**.** X 上の根付き二分系統ネットワーク N が与えられたとき， N の全域系統樹 T が存在するか否かを決定せよ．存在するならば，N の全域系統樹 T を一つ見つけよ．

問題 1 が入力サイズ $|A(N)|$ に関する線形時間で解けることを示した先行研究は幾つかあるが（[7, 21]），本質的に重要なのは全域系統樹を作るうえで「許容可能（admissible）」なアークの選び方を定める定義 4.1 と，その必要十分性を示す定理 4.2 である．

定義 4.1 ([7])**.** N を X 上の根付き二分系統ネットワークとする． $A(N)$ の部分集合 S が *admissible* であるとは， S が以下の条件を全て満たすことをいう:

C0　S は $deg^-_N(v) = 1$ または $deg^+_N(u) = 1$ を満たす任意の $(u, v) \in A(N)$ を含む

C1　$head(a_1) = head(a_2)$ を満たす任意の $a_1, a_2 \in A(N)$ に対し，$\{a_1, a_2\}$ の要素のうちちょうど 1 つが S に含まれる．

C2　$tail(a_1) = tail(a_2)$ を満たす任意の $a_1, a_2 \in A(N)$ に対し，$\{a_1, a_2\}$ の要素のうち少なくとも 1 つが S に含まれる．

定理 4.2 ([7])**.** X 上の根付き二分系統ネットワーク N に対して，N の任意の全域系統樹

は，$A(N)$ の *admissible* な部分集合 S が誘導する N の部分グラフ $N[S]$ である．さらに，$A(N)$ の *admissible* な部分集合 S たちの族と，N の全域系統樹のアーク集合の族は一対一に対応する．

ところで，任意の根付き二分系統ネットワークは，新たに幾つかのリーフを追加することで tree-based network に変換することができることが知られている（図 5）．

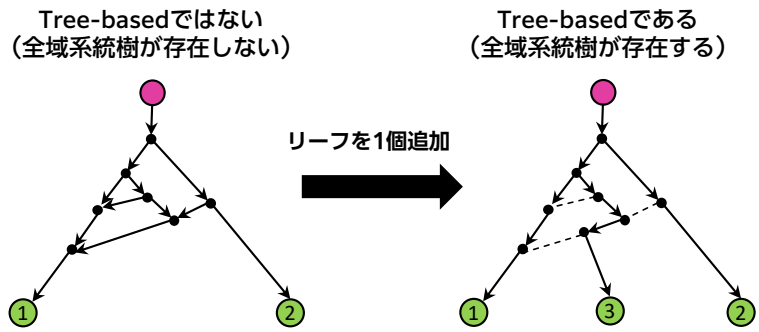

図 5　リーフを追加することで全域系統樹を持たない系統ネットワークを tree-based network に変換できる．この例では 1 個のリーフを追加するだけで変換できた．

この観察から，問題 1 の亜種といえる，次の**逸脱度評価問題**というものが現れる．論文 [6] では問題 2 が $O(|V(N)|^{3/2})$ 時間で解けることが示されていた．

問題 2 ([6])**.** X 上の根付き二分系統ネットワーク N が与えられたとき， N を tree-based network に変換するために必要な追加リーフ数の最小値 $\delta(N)$（≥ 0）を求めよ．

決定問題が解の個数が 0 か否かだけを尋ねるのに対して**数え上げ問題**は解の個数を明示的に尋ねる問題である．その一つである次の問題 3 も論文 [7] で言及されて以来，応用上重要だが時間計算量が自明でない計算問題として注目を集めた．その後の論文 [16] で問題 3 が多項式時間で解けることは示されたが，具体的な計算量は知られていなかった．

問題 3 ([7])**.** X 上の根付き二分系統ネットワーク N が与えられたとき， N の全域系統樹の個数 $\alpha(N)$ を決定せよ．

次の問題 4 は，数え上げの発展型といえる**列挙問題**である．論文 [7] では $\alpha(N) = 2^r$（r は N に含まれる合流点の個数）となる系統ネットワーク N が存在する（つまり全域系統樹を全列挙するのに要する時間は入力サイズの指数時間になりうる）ことが指摘されていた．

問題 4 ([7])**.** X 上の根付き二分系統ネットワーク N が与えられたとき， N の全域系統樹 $T_1, \ldots, T_{\alpha(N)}$ を全て列挙せよ．

この問題 4 に限らず， 一般に，列挙問題の数の個数は入力サイズの多項式関数で抑え

られるとは限らず，場合によっては解が無数に存在することもある．そのため，通常の計算問題とは異なり，列挙アルゴリズムの効率の良さを評価する際には，入力サイズだけでなく出力サイズも考慮しなければならない．列挙アルゴリズムの中で効率的なクラスの一つとされるのは**多項式時間遅延アルゴリズム** [14] で，これは一つの解を見つけてから次の解を見つけるまでの時間が毎回入力サイズに関する多項式関数で抑えられるようなやり方で全ての解を次々に列挙するものである [8]．これに該当するアルゴリズムは実行時間が出力サイズに対して線形となるので，列挙するべきものの個数が増大しても計算時間が膨れ上がらないという点で確かに優れている．よって，問題 4 に対する多項式時間遅延アルゴリズムが存在すれば，問題 4 は効率的に解ける問題であるといえる．

Tree-based network の各アークを非負実数値（例：確率）で重み付けすると，全域系統樹に関する**最適化問題**（例：尤度最大の全域系統樹を求める問題）が自然に現れることは既に図 2 で見た通りだが，少々意外なことに講演者の論文 [10] は重み付きの tree-based network を扱った最初の研究であり，したがって問題 5 に関する先行研究も当然なかった．N の全域系統樹の総数 $\alpha(N)$ が入力サイズに関して指数関数的になりうることを思い出せば，それら全てに対して目的関数の値を計算して最適なものを選ぶという方法では指数時間を要することがすぐに分かるが，多項式時間で解くアルゴリズムが存在するか否かは全く自明ではない．

問題 5 ([10])**.** X 上の根付き二分系統ネットワーク N とそれに関する重み付け関数 $w\colon A(N)\to\mathbb{R}_{\geq 0}$ が与えられたとき，目的関数 $f(T)=\sum_{a\in A(T)} w(a)$ の値を最大化する N の全域系統樹 T を求めよ [*4].

5 根付き二分系統ネットワークの構造定理

前節では系統ネットワークと全域系統樹に関する一連の計算問題とそれぞれの先行研究について概説した．この節ではそれらの多様な問題を統一的なアプローチで解くことを可能にする「系統ネットワークの構造定理」を紹介する．まずは，この定理を述べるために必要な用語を導入する．

X 上の根付き二分系統ネットワーク N に対して，N **のジグザグ・トレイル**とは，$|A(Z)|\geq 1$ を満たす N の連結部分グラフ Z で，任意の $i\in[1,m-1]$ に対して $head(a_i)=head(a_{i+1})$ or $tail(a_i)=tail(a_{i+1})$ であるような $A(Z)$ の置換 $(a_1,\ldots,a_m)$ のことである．N の任意のジグザグ・トレイル Z は，（必ずしも互いに異なるとは限らない）頂点と，互いに異なるアークの交互列で表すことができる（例えば $(v_0,(v_0,v_1),v_1,(v_2,v_1),v_2,(v_2,v_3),\ldots,(v_m,v_{m-1}),v_m)$）．しかし，本稿ではより簡潔に，上

[*4] ここでは各アークの不確かさに応じた確率 w が与えられている状況で尤度または対数尤度 $f(T)$ が最大となる N の全域系統樹 T を求めるという最尤推定の文脈を想定しているが，もちろん確率以外の重み（時間や長さ）を考えても良いし，符号を変えて最小化問題にしても良いし，目的関数を和ではなく積の形 $f(T)=\prod_{a\in A(T)} w(a)$ に設定しても良い（log をとれば同じことである）．

記の Z を $v_0 > v_1 < v_2 > v_3 < \cdots > v_{m-1} < v_m$（またはその reverse order）という記法で表す．混乱の恐れがない場合は，$(a_1, \ldots, a_m)$ という表記も用いる．

N のジグザグ・トレイル Z が**極大**であるとは，Z が Z' の真部分グラフであるような N のジグザグ・トレイル Z' が存在しないことをいう．N の任意のジグザグ・トレイル Z は次の 4 つのいずれかに分類できる（図 6）．N のジグザグ・トレイル Z は，$m := |A(Z)| \geq 4$ が偶数で, Z が $v_0 < v_1 > v_2 < v_3 > \cdots > v_{m-2} < v_{m-1} > v_m = v_0$ の形で表せるときに**クラウン**といい，そうでない場合，Z は**フェンス**という．さらに，$m := |A(Z)| \geq 1$ が奇数であるフェンス Z は N-**フェンス**といい，$Z : v_0 > v_1 < v_2 > v_3 < \cdots > v_{m-2} < v_{m-1} > v_m$ の形で表される．$m \geq 2$ が偶数であるフェンス Z は $Z : v_0 > v_1 < v_2 > v_3 < \cdots < v_{m-2} > v_{m-1} < v_m$ の形で表されるとき W-**フェンス**といい，$Z : v_0 < v_1 > v_2 < v_3 > \cdots > v_{m-2} < v_{m-1} > v_m$ の形で表されるとき M-**フェンス**という. 任意のフェンス Z に対して, 両端の二頂点 v_0 と v_m は N の *endpoints* という．

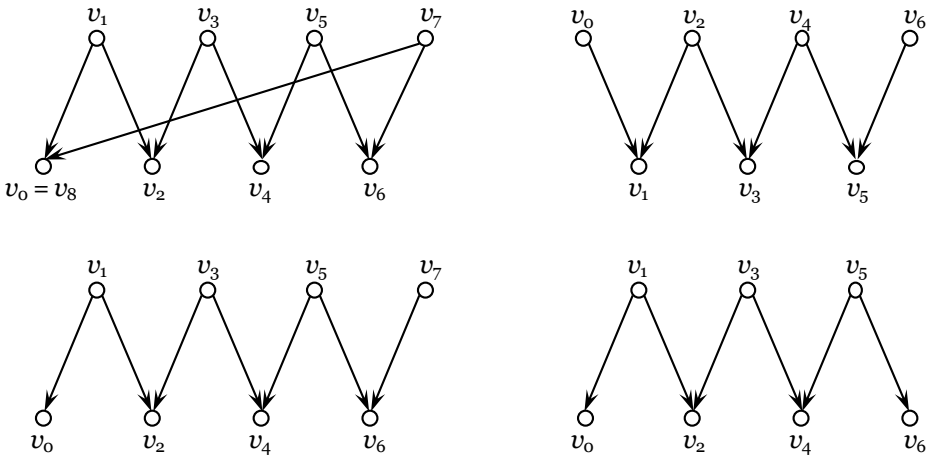

図 6　N における極大ジグザグ・トレイル Z の 4 つのケース．左上：クラウン．左下：N-フェンス．右上：W-フェンス．右下：M-フェンス．

注 5.1. フェンスとクラウンという用語は poset の理論において使われるものであり，通常は図 6 に示すように二部グラフの形で表されるものを指すが [17]．本稿では図 7 のような非典型的な M-フェンスを認めていることに注意されたい．

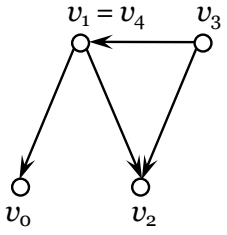

図 7　注 5.1 の例．これは $v_0 < v_1 > v_2 < v_3 > v_4$ の形で表される M-フェンスである．

定理 5.2 (根付き二分系統ネットワークの構造定理 (1)). X 上の任意の根付き二分系統ネットワーク N に対して，N の分解 $\mathcal{Z} = \{Z_1, \ldots, Z_\ell\}$ で，各 $Z_i \in \mathcal{Z}$ が N のジグザグ・トレイルであるようなものは，唯一つ定まる．

定理 5.2 で言及している $\mathcal{Z}$ を，N **の極大ジグザグ・トレイル分解**と呼ぶ．

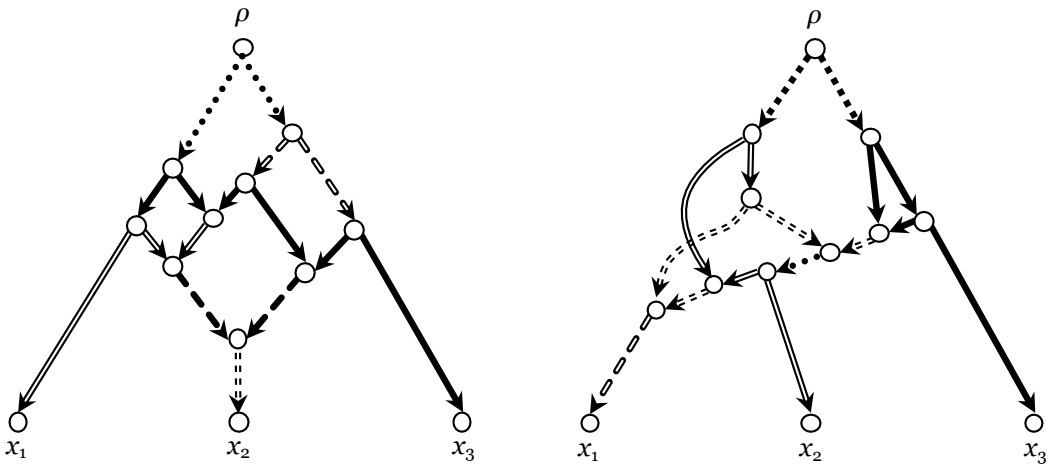

図 8　定理 5.2 の例．$X = \{x_1, x_2, x_3\}$ 上の根付き二分系統ネットワーク N の極大ジグザグ・トレイル分解．複数の種類の線を用いたのは，異なる極大ジグザグ・トレイル s を強調表示するためである．いずれのネットワークも，3 つの極大 M-フェンス，2 つの極大 N-フェンス，1 つの極大 W-フェンスに分解される. 右は文献 [7] で論じられたネットワークを題材とした例である．なお，図 7 の M-フェンスと同型であるものが黒の太線矢印で描かれている．

補題 5.3. N は X 上の任意の根付き二分系統ネットワークで，$\mathcal{Z} = \{Z_1, \ldots, Z_\ell\}$ は N の極大ジグザグ・トレイル分解とする．このとき，$S \subseteq A(N)$ が $A(N)$ の *admissible* な部分集合であることと，各 $i \in [1, \ell]$ に対して $S_i := S \cap A(Z_i)$ が $A(Z_i)$ の *admissible* な部分集合であることは同値である．

補題 5.3 の証明は，S が定義 4.1 の条件 C0，C1，C2 を満たすことと，各 $i \in [1, \ell]$ に対して次の C0′，C1′，C2′ が成立することが同値であることを示すことによる：

C0′　S は $deg^-_{Z_i}(v) = 1$ または $deg^+_{Z_i}(u) = 1$ を満たす任意の $(u, v) \in A(Z_i)$ を含む

C1′　$head(a_1) = head(a_2)$ を満たす任意の $a_1, a_2 \in A(Z_i)$ に対し，$\{a_1, a_2\}$ の要素のうちちょうど 1 つが S_i に含まれる．

C2′　$tail(a_1) = tail(a_2)$ を満たす任意の $a_1, a_2 \in A(Z_i)$ に対し，$\{a_1, a_2\}$ の要素のうち少なくとも 1 つが S_i に含まれる．

以後，N の極大ジグザグ・トレイル s の順序を固定し，$\mathcal{Z}$ の代わりに列 $(Z_1, \ldots Z_\ell)$ を考える．これにより，N の部分グラフ G を直積 $\prod_{i \in [1,\ell]} A(G) \cap A(Z_i)$ と同一視することがで

きるようになる．さらに，各 $i \in [1,\ell]$ に対して，$A(Z_i)$ の要素の順序も固定し，$A(Z_i)$ の代わりに列 $(a_1,\ldots,a_{m_i})$ を考える（$m_i := |A(Z_i)|$）．これにより，$A(Z_i)$ の任意の部分集合を長さ m_i の 0-1 配列でエンコードできるようになる．例えば，$A(Z_i) = (a_1,a_2,a_3,a_4,a_5)$ について，部分集合 $\{a_2,a_4,a_5\} \subseteq A(Z_i)$ は $\langle 0\,1\,0\,1\,1\rangle = \langle (01)^2 1\rangle$ で表すことができる．この記法を用いて，各 $i \in [1,\ell]$ につき，$A(Z_i)$ の部分集合の族 $\mathscr{S}(Z_i)$ を次のように定義する．

$$\mathscr{S}(Z_i) := \begin{cases} \{\langle (10)^{m_i/2}\rangle, \langle (01)^{m_i/2}\rangle\} & (Z_i \text{ がクラウンの場合}) \\ \{\langle 1(01)^{(m_i-1)/2}\rangle\} & (Z_i \text{ が N-フェンスの場合}) \\ \{\langle 1(01)^p(10)^q 1\rangle \mid p,q \in \mathbb{Z}_{\geq 0}, p+q=(m_i-2)/2\} & (Z_i \text{ が M-フェンスの場合}) \end{cases} \tag{1}$$

なお，上の 0-1 配列を用いた表現は，その対称性より，順列 $(a_1,\ldots,a_{m_i})$ の決め方に依らない．例えば Z_i が N-フェンスの場合，$\langle 1(01)^{(m_i-1)/2}\rangle$ とその reverse ordering は同一である．

定理 5.4 (根付き二分系統ネットワークの構造定理 (2)). N は X 上の任意の根付き二分系統ネットワークで，$\mathcal{Z} = \{Z_1,\ldots,Z_\ell\}$ は N の極大ジグザグ・トレイル分解とする．このとき，N が全域系統樹を持つことと，$\mathcal{Z}$ のどの要素 Z_i も W-フェンスでないことは同値である．また，N が全域系統樹を持つとき，N の全域系統樹の集まり $\mathcal{T}$ は，式 (1) の集合族 $\mathscr{S}(Z_i)$ を用いて次のように特徴づけられる：

$$\mathcal{T} = \prod_{i\in[1,\ell]} \mathscr{S}(Z_i). \tag{2}$$

6 問題 1–5 に対する線形時間・線形時間遅延アルゴリズムの導出

定理 5.4 から，これまでに述べた一連の計算問題に対する高速なアルゴリズムを直ちに導かれる．どのアルゴリズムも N の極大ジグザグ・トレイル分解 $\mathcal{Z}$ を計算するという前処理からスタートするが，N の極大ジグザグ・トレイル分解 $\mathcal{Z}$ は N の各アークを 1 回ずつ通るだけで得られるから，この前処理は $O(|A(N)|)$ 時間でできる．入力 N の読み込みには $\Omega(N)$ 時間を要するから，次の命題の通り，この前処理は $\Theta(|A(N)|)$ 時間でできる．

命題 6.1. X 上の任意の二分系統ネットワーク N に対して，N の極大ジグザグ・トレイル分解 $\mathcal{Z} = \{Z_1,\ldots,Z_\ell\}$ は $\Theta(|A(N)|)$ 時間で計算できる．

定理 5.4 の前半部分と命題 6.1 より，問題 1 の決定問題については，N の極大ジグザグ・トレイル分解 $\mathcal{Z}$ に W-フェンスがあるか否かをチェックするという $\Theta(|A(N)|)$ 時間アルゴリズムが得られる．探索問題については，N の各ジグザグ・トレイル Z_i に対応する $\mathscr{S}(Z_i)$ から任意の要素を一つだけ選択することで全域系統樹を誘導するアーク集合が得られるため，やはり $\Theta(|A(N)|)$ 時間で解ける．決定問題の発展型である問題 2 については，$\mathcal{Z}$ に含まれる W-フェンスの個数を数えるという $\Theta(|A(N)|)$ 時間アルゴリズムが得ら

れる．何故なら，N の極大 W-フェンスはリーフを一つ追加するだけで解消することができるので（図 9），問題 2 で要求されている $\delta(N)$ は N の極大ジグザグ・トレイル分解 $\mathcal{Z}$ に含まれる W-フェンスの個数と等しいからである（当然，その個数は $\Theta(|A(N)|)$ 時間で計算できる）．

定理 5.4 によって N の全域系統樹の集まり $\mathcal{T}$ が式 (1) の集合族 $\mathscr{S}(Z_i)$ の直積で表されたので，系 6.2 の通り，全域系統樹の個数 $\alpha(N) = |\mathcal{T}|$ は各極大ジグザグ・トレイル Z_i における $A(Z_i)$ の admissible な部分集合の個数 $\alpha(Z_i) = |\mathscr{S}(Z_i)|$ の積と等しい．したがって問題 3 も $\Theta(|A(N)|)$ 時間で解ける．

系 6.2. *N は $\alpha(N) \in \mathbb{Z}_{\geq 0}$ 個の全域系統樹を持つ X 上の二分系統ネットワークで，$\mathcal{Z} = \{Z_1, \ldots, Z_\ell\}$ は N の極大ジグザグ・トレイル分解とする．このとき，$\alpha(Z_i)$ は式 (3) のようになり，$\alpha(N) = \prod_{i=1}^{\ell} \alpha(Z_i)$ が成立する．*

$$\alpha(Z_i) = \begin{cases} 0 & (Z_i \text{ が } W\text{-フェンスの場合}) \\ 1 & (Z_i \text{ が } N\text{-フェンスの場合}) \\ 2 & (Z_i \text{ がクラウンの場合}) \\ |A(Z_i)|/2 & (Z_i \text{ が } M\text{-フェンスの場合}) \end{cases} \tag{3}$$

探索問題を解く前述の線形時間アルゴリズムで全域系統樹を一つ得たら，その全域系統樹を構成する $\mathscr{S}(Z_i)$ の要素のどれかを別の要素に入れ替えれば新しい組合せが作れるので，毎回線形時間で次々に全域系統樹を列挙できる．全域系統樹の個数 $\alpha(N)$ が線形時間で計算できるので，全ての要素を列挙し終えたか否かの判定も線形時間でできる．よって，問題 4 も線形時間遅延で解くことができる．問題 5 についても，最適な全域系統樹のアーク集合が各 $\mathscr{S}(Z_i)$ の中で最適な要素を集めたものであることに気づけば，$\Theta(|A(N)|)$ 時間で解けることが直ちに分かる．

図 9　問題 2 に対するアルゴリズムの要点（詳細は本文を参照）．

7　全域系統樹の個数 $\alpha(N)$ を数える意義

系統ネットワークの構造定理から導かれたアルゴリズムによって何ができるようになったのかを，数え上げの具体例を通じて見てみよう．ある生物学者が $x_1, \ldots, x_8$ という 8 つの種に関して考えられる進化の道筋を全て描き出して一つの図にまとめ上げたところ，

系統樹とは似ても似つかない，図 10 の複雑な系統ネットワーク N が得られたとする．このようなネットワークは生物学的に意味のある情報を何も持っていないように見えるが，果たして本当にそうだろうか？全域系統樹の個数 $\alpha(N)$ を数えることで，それを考察してみよう．N を極大ジグザグ・トレイル分解すると，21 個の極大 N-フェンスと 7 個の 極大 M-フェンスになるので，全域系統樹の個数は $\alpha(N) = 7! = 5040$ である [*5]．$\alpha(N) = 5040$ という数字は大きいように感じられるかもしれないが，$X = \{x_1, \ldots, x_8\}$ 上の根付き二分系統樹の個数 $(2|X|-3)!! = 13 \times 11 \times \cdots \times 5 \times 3 \times 1 = 135135$ に比べれば少ない．よって，この N には興味のある現存種の集合 X 上の全ての系統樹を網羅するほど無秩序で乱雑なものではなく，真の系統樹の候補を（それなりに）絞り込んでくれているといえる．このように，サイズの大きな N に対しても容易に計算できる全域系統樹の個数 $\alpha(N)$ は，系統ネットワーク N の乱雑さを表す簡便な指標として利用することができる．

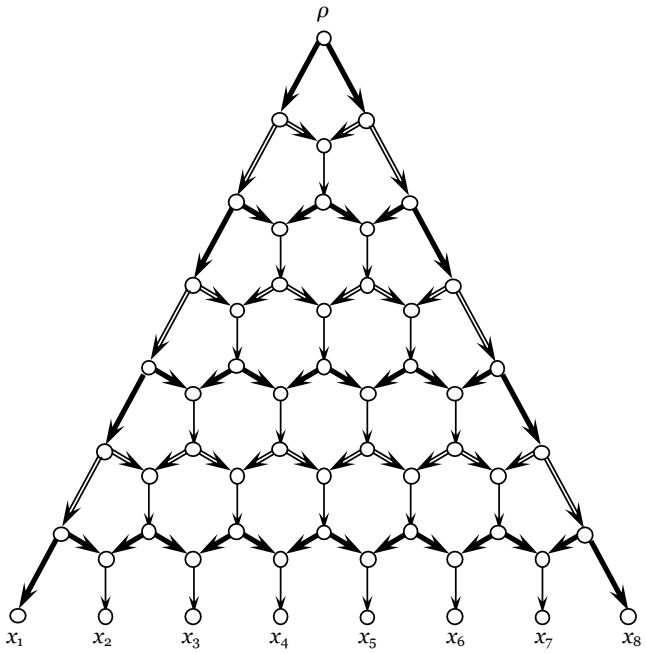

図 10　数え上げの応用を説明するために用いた系統ネットワークの例（詳細は本文を参照）．

8　おわりに

系統ネットワークに含まれる系統樹を探るという問題意識に基づく組合せ論の問題は色々なものがあるが（例えば [20, 22, 15, 4]）．本稿では特に根付き二分系統樹が全域木の形で根付き二分系統ネットワークに含まれている状況を考え，これに関する最も基本的な計算問題を 5 つ紹介した．ここでは紹介しなかったが，牧野氏との共著論文 [11] では，列挙と最適化の発展型でありバイオインフォマティクス的な意義もある，全域系統樹の

[*5] この値を $\alpha(N)$ の自明な upper bound $2^r = 2^{21} = 2097152$（r は N の合流点の数）と比較すると，$\alpha(N)$ の厳密な個数を数える意義が分かる．

「上位ランキング問題」（問題 6）に対して線形時間遅延アルゴリズムを与えた．一般に，任意の k 個を列挙する問題と上位 k 個を順に列挙する問題では後者のほうが難しいはずだが，全域系統樹に関しては列挙問題と上位ランキング問題の時間計算量が同程度というのは興味深い．

問題 6 ([11])**.** X 上の根付き二分系統ネットワーク N，それに関する重み付け関数 $w\colon A(N)\to\mathbb{R}_{\geq 0}$，および $k\in\mathbb{Z}_{\geq 0}$ が与えられたとき，目的関数 $f(T)=\sum_{a\in A(T)}w(a)$ の値の最大を与えるものから順に N の全域系統樹を k 個列挙せよ．

今回紹介した tree-based network と全域系統樹というテーマに限っても，組合せ論的系統学にはまだまだ多数の興味深い問題がある．本稿で解説した全域系統樹の数え上げと密接に関連するが未だに計算量について何も知られていない問題として，論文 [10] では同型なものを除く全域系統樹の数え上げと，位相同型なものを除く全域系統樹の数え上げという二つの問題を挙げたが，最近はこれらの問題をはじめとする諸問題を学生と共に研究し，まだ部分的ではあるが色々な進展が得られている．組合せ論的系統学は生物学の知識の有無を問わず参入できるため，様々なバックグラウンドの研究者を巻き込んで成長している分野だが，日本ではほとんどの方にとってまだ馴染みがないかもしれない．この特別講演を通じて，この研究分野に少しでも興味を持っていただけたら幸いである．

参考文献

[1] M. Anaya, O. Anipchenko-Ulaj, A. Ashfaq, J. Chiu, M. Kaiser, M. S. Ohsawa, M. Owen, E. Pavlechko, K. St. John, S. Suleria, K. Thompson, and C. Yap, *On determining if tree-based networks contain fixed trees*, Bulletin of mathematical biology **78** (2016), no. 5, 961–969.

[2] M. Bordewich and C. Semple, *A universal tree-based network with the minimum number of reticulations*, Discrete Applied Mathematics **250** (2018), 357–362.

[3] J. E. Cohen, *Mathematics is biology's next microscope, only better; biology is mathematics' next physics, only better*, PLoS biology **2** (2004), no. 12, e439.

[4] N. Davidov, A. Hernandez, J. Jian, P. McKenna, K.A. Medlin, R. Mojumder, M. Owen, A. Quijano, A. Rodriguez, K. St John, et al., *Maximum covering subtrees for phylogenetic networks*, IEEE/ACM Transactions on Computational Biology and Bioinformatics **18** (2020), no. 6, 2823–2827.

[5] W. F. Doolittle, *Phylogenetic classification and the universal tree*, Science **284** (1999), no. 5423, 2124–2128.

[6] A. Francis, C. Semple, and M. Steel, *New characterisations of tree-based networks and proximity measures*, Advances in Applied Mathematics **93** (2018), 93–107.

[7] A. R. Francis and M. Steel, *Which phylogenetic networks are merely trees with additional arcs?*, Systematic Biology **64** (2015), no. 5, 768–777.

[8] L. A. Goldberg, *Efficient algorithms for listing combinatorial structures*, vol. 5, Cambridge University Press, 2009.

[9] M. Hayamizu, *On the existence of infinitely many universal tree-based networks*, Journal of

Theoretical Biology **396** (2016), 204–206.

[10] ———, *A structure theorem for rooted binary phylogenetic networks and its implications for tree-based networks*, SIAM Journal on Discrete Mathematics **35** (2021), no. 4, 2490–2516.

[11] M. Hayamizu and K. Makino, *Ranking top-k trees in tree-based phylogenetic networks*, (2022), preprint.

[12] D. H. Huson, R. Rupp, and C. Scornavacca, *Phylogenetic networks: concepts, algorithms and applications*, Cambridge University Press, 2010.

[13] L. Jetten and L. van Iersel, *Nonbinary tree-based phylogenetic networks*, IEEE/ACM transactions on computational biology and bioinformatics **15** (2016), no. 1, 205–217.

[14] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, *On generating all maximal independent sets*, Information Processing Letters **27** (1988), no. 3, 119–123.

[15] S. Linz, K. St. John, and C. Semple, *Counting trees in a phylogenetic network is #P-complete*, SIAM Journal on Computing **42** (2013), no. 4, 1768–1776.

[16] J. C. Pons, C. Semple, and M. Steel, *Tree-based networks: characterisations, metrics, and support trees*, Journal of Mathematical Biology (2018).

[17] B. Schröder, *Ordered sets: An introduction with connections from combinatorics to topology 2nd ed.*, Birkhäuser, 2016.

[18] C. Semple and M. Steel, *Phylogenetics*, Oxford Lecture Series in Mathematics and its Applications, vol. 24, Oxford University Press, Oxford, 2003. MR 2060009 (2005g:92024)

[19] M. Steel, *Phylogeny: Discrete and random processes in evolution*, SIAM, 2016.

[20] L. van Iersel, C. Semple, and M. Steel, *Locating a tree in a phylogenetic network*, Information Processing Letters **110** (2010), no. 23, 1037–1043.

[21] L. Zhang, *On tree-based phylogenetic networks*, Journal of Computational Biology **23** (2016), no. 7, 553–565.

[22] J. Zhu, Y. Yu, and L. Nakhleh, *In the light of deep coalescence: revisiting trees within networks*, BMC bioinformatics **17** (2016), no. 14, 271–282.