Boltzmann Sampler を用いた重み付きカクタスグラフのランダムサンプリング

鈴木 尭虎 [1], 小堤 幹太 [2], 早水 桃子 [3]
[1] 早稲田大学大学院 基幹理工学研究科，[2] 早稲田大学 基幹理工学部，[3] 早稲田大学 理工学術院
e-mail : takatora.szk@fuji.waseda.jp

1　概要

離散構造の一様ランダムサンプリングは，アルゴリズムの性能評価や統計的推論に不可欠であり，Duchon ら [1] は「Boltzmann sampler」という高速サンプリングのための枠組みを提案した．我々は，この手法をツリーとサイクルのみで構成されるカクタスグラフとその距離空間に適用した．本講演では，Boltzmann sampler を用いてカクタスグラフを一様ランダムに生成するアルゴリズム [2] を発展させて，整数重み付きカクタスグラフを一様ランダムに生成するアルゴリズムを提案する．これは，カクタスグラフを最適な実現に持つ有限距離空間の一様ランダム生成と同値である．

2　距離空間の実現

ある有限距離空間 (X, d) に対して，任意の $x, y \in X$ に対して $d(x, y) = d_G(x, y)$ が成立するような重みつきグラフ G を，(X, d) の**実現**という．ここで，d_G は G 上の最短経路距離を表す．G が (X, d) の実現であり，かつ G からどの辺を取り除いても (X, d) の実現でなくなるとき，G を *minimal* **な実現**という．さらに，(X, d) の全ての実現の中で全長（辺重みの総和）が最小となるような実現を**最適な実現**という．

一般の有限距離空間に対して，その最適な実現は一意とは限らない上に高々可算個とも限らない [3]．しかし，いくつかのグラフのクラスに対しては最適な実現の一意性が保証されることもある．たとえば木 T を実現にもつ距離空間（木距離）であれば，T が最適かつ一意な実現となることが知られている [4]．また，Imrich [5] によって，サイクルを最適かつ一意な実現に持つ距離空間の特徴づけが以下のように与えられた．

定理 1（Imrich (2004)[5], Theorem 4.4） ある有限距離空間 (X, d) がサイズ 4 以上のサイクル C を minimal な実現としてもつとし，C の頂点を $\{v_i \mid 1 \leq i \leq n\}$，$C$ の辺を $\{(v_i, v_{i+1}) \mid 1 \leq i \leq n-1\} \cup \{(v_n, v_1)\}$ とする．このとき，C が最適かつ一意な実現である必要十分条件は，任意の i で

$$d(v_i, v_{i+1}) + d(v_{i+1}, v_{i+2}) = d(v_i, v_{i+2}) \tag{1}$$

が n を法として成り立つことである．

さらに，この定理 1 を用いて，距離空間 (X, d) がカクタスグラフを実現にもつならば，(X, d) の最適な実現はカクタスグラフであり，一意に定まることが早水ら [6] によって示された．

つまり，全長をそれ以上短縮できないカクタスグラフと，カクタスで実現できる距離空間（カクタス距離空間）の間には一対一の対応がある．よって，そのようなカクタスグラフを生成することで，カクタス距離空間を生成できる．

3 Boltzmann sampler

$\mathcal{C}$ を高々可算な集合とする．$\mathcal{C}$ 上の関数 $|\cdot| : \mathcal{C} \to \mathbb{Z}_{\geq 0}$ であって，任意の $n \in \mathbb{Z}_{\geq 0}$ について $|\gamma| = n$ となる $\gamma \in \mathcal{C}$ の個数が有限であるとき，関数 $|\cdot|$ を**サイズ関数**とよび，$(\mathcal{C}, |\cdot|)$ のことを**クラス**とよぶ．クラス $(\mathcal{C}, |\cdot|)$ の母関数を $C(z) := \sum_{\gamma \in \mathcal{C}} z^{|\gamma|}$ と定める．ここで，x を $C(z)$ の収束半径内にある任意の正数として固定したとき，構造 $\gamma \in \mathcal{C}$ を生成する確率が $\mathbb{P}(\gamma) = x^{|\gamma|}/C(x)$ で与えられるサンプリングアルゴリズムを *Boltzmann sampler* という．この定義から，構造 γ が生成される確率は常に γ のサイズのみに依存するため，同じサイズの離散構造は一様ランダムに生成される．

ここで，特定の離散構造を生成する Boltzmann sampler を構成する際には，目的の離散構造を既知の単純なクラスを組み合わせた形式で表現する必要がある．この基礎となる単純なクラスはいくつかあるが，紙面の都合上本稿では省略する（詳細は文献 [1] を参照）．Panagiotou ら [2] はカクタスグラフを特徴づけるクラスの構成方法と，一様ランダムに生成するアルゴリズムを与えている．

4 重み付きカクタスグラフのランダムサンプリング

本講演では，Boltzmann sampler を用いて重み付きカクタスグラフとカクタス距離空間を高速に一様ランダムに生成するアルゴリズムを提案する．提案手法の流れは以下の通りである．

1) 重みのないカクタスグラフをランダムに生成する Boltzmann sampler [2] によってカクタスグラフ G をサンプリングする（このとき G に含まれるサイクルは長さ 4 以上に制限する）．
2) n を定数，G の辺の本数を m とする．n の m 個への整数分割を生成する Boltzmann sampler によって分割をサンプリングし，これを用いて G の各辺に重みを与える．
3) 式 (1) の制約を満たしているか確認する．
4) 重み付きカクタスグラフ G を出力する．

提案手法を実装して計算機上で実行した結果と計算時間に関しては，発表時に紹介する．

謝辞 本研究は，JST 次世代研究者挑戦的研究プログラム JPMJSP2128 の支援を受けたものです．

参考文献

[1] P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13(4–5):577–625, 2004.

[2] K. Panagiotou, L. Ramzews, and B. Stufler. Exact-size sampling of enriched trees in linear time. *SIAM Journal on Computing*, 52(5):1097–1131, 2023.

[3] I. Althöfer. On the complexity of searching game trees and other recursion trees. *Journal of Algorithms*, 9(4):538–567, 1988.

[4] S. L. Hakimi and S. S. Yau. Distance matrix of a graph and its realizability. *Quarterly of applied mathematics*, 22(4):305–317, 1965.

[5] W. Imrich, J. M. S. Simões-Pereira, and C. M. Zamfirescu. On optimal embeddings of metrics in graphs. *Journal of Combinatorial Theory, Series B*, 36(1):1–15, 1984.

[6] M. Hayamizu, K. T. Huber, V. Moulton, and Y. Murakami. Recognizing and realizing cactus metrics. *Information Processing Letters*, 157:105916, 2020.

Tree-child network に向きづけ可能かを効率的に判定できる無向グラフのクラスについて

前田 隼佑 [1], 浦田 剛志 [1], 早水 桃子 [2]
[1] 早稲田大学大学院 基幹理工学研究科 数学応用数理専攻, [2] 早稲田大学理工学術院
e-mail : shunsuke.m0131@moegi.waseda.jp

1　概要

生物系統学の研究分野では距離行列（非類似度）から系統ネットワークを作る手法が普及しているが，距離に基づいて推定できるのは無向グラフであるため，進化の流れや道筋を直ちに解釈できない．そこで近年では無向系統ネットワークの辺を適切に向きづけして有向系統ネットワークに変換する向きづけ問題が論じられるようになった．例えば，与えられた無向グラフが tree-based network に向きづけ可能かを判定する問題は NP 完全である [1]．一方，無向グラフ G が tree-based network に向きづけ可能であることと，G の cherry cover graph が存在することは同値である [2]．他方，Tree-child network に向きづけ可能かを判定する問題（Tree-Child Orientation 問題）には実用的な FPT アルゴリズムやヒューリスティックスが提案されているが [3]，この問題も NP 完全と予想され [4]，頂点の最大次数が 5 の場合に限り NP 完全性が証明されている [5].

本講演では，cherry cover graph に着目して，tree-child network に向きづけ可能なグラフの性質を考察した結果を述べる．

2　準備

文献 [4] と同様の表記を用いる．有限集合 X を生物の現存種を表し，**リーフ集合**とよび，X の要素を**リーフ**という．無向グラフ N が，次数 2 の頂点がなく，次数が 1 の頂点集合が X と一致するとき，N を**無向系統ネットワーク**という．また，リーフ以外の全頂点の次数が 3 のときは N は**二分**であるという．本講演で扱うすべての有向グラフは非巡回なグラフ（DAG）である．

定義 1　X 上の**根付き有向系統ネットワーク** $\widehat{N}$ は，有向非巡回グラフのうち，以下の 3 つの条件を満たすものである．

- $indeg_{\widehat{N}}(\rho) = 0$，$outdeg_{\widehat{N}}(\rho) = 2$ となる頂点 $\rho \in V(\widehat{N})$ がただ 1 つ存在する．
- $\widehat{N}$ の $indeg_{\widehat{N}}(x) = 1$，$outdeg_{\widehat{N}}(x) = 0$ を満たす頂点 x の集合は，X と一致する．
- ρ 以外の頂点の次数は 2 ではない．

ここで，入次数 0 の頂点 ρ を**根**とよび，入次数 1 の頂点を **tree vertex**，入次数が 2 以上の頂点を **reticulation** という．

また，根とリーフ以外のすべての頂点の次数が 3 であるとき，$\widehat{N}$ は**二分**であるという．

3　Tree-child network

X 上の根付き二分有向系統ネットワーク $\widehat{N}$ のリーフでない頂点の子が少なくとも 1 つはリーフか tree vertex であるとき，$\widehat{N}$ は，**tree-child network** であるという．Tree-child network の特徴を容易に理解するために以下の定理を記す．

定理 2（[6], Lemma2）　有向系統ネットワーク $\widehat{N}$ が tree-child network である必要十分条件は，$\widehat{N}$

のすべての頂点について，以下の 2 つの条件を満たすものである．

- 子に reticulation を持つような reticulation はない．
- 子である 2 つの頂点がどちらも reticulation であるような tree vertex は存在しない．

4 Tree-Child Network に向きづけ可能なグラフの性質

N のある 1 つの辺 e_ρ に根となる頂点 ρ を挿入して N_ρ を作る操作を N の *rooting* という．また，N に rooting した後，さらに N_ρ の各辺に向きを割り当てる操作を N の *orienting* という．

問題 3 (**Tree-Child Orientation**)

INPUT： 二分無向系統ネットワーク N

OUTPUT： N が tree-child ネットワークに向きづけ可能であるならば，N に向きづけをした tree-child ネットワーク，そうでないならば NO.

本講演では，系統ネットワークの向きづけ問題に関連する研究に言及した上で，tree-child network に向きづけ可能なグラフが持つ性質を，グラフの cherry cover に着目して論じる．

謝辞 本研究の一部は，JST 次世代研究者挑戦的研究プログラム JPMJSP2128 と早稲田大学理工学術院総合研究所の若手研究者支援事業（アーリーバードプログラム）の支援を受けたものである．

参考文献

[1] Katharina T Huber, Leo van Iersel, Remie Janssen, Mark Jones, Vincent Moulton, Yukihiro Murakami, and Charles Semple. Orienting undirected phylogenetic networks. *Journal of Computer and System Sciences*, 140:103480, 2024.

[2] Leo van Iersel, Remie Janssen, Mark Jones, Yukihiro Murakami, and Norbert Zeh. A unifying characterization of tree-based networks and orchard networks using cherry covers. *Advances in Applied Mathematics*, 129:102222, 2021.

[3] Tsuyoshi Urata, Manato Yokoyama, and Momoko Hayamizu. Orientability of undirected phylogenetic networks to a desired class: Practical algorithms and application to tree-child orientation. *arXiv preprint arXiv:2407.09776*, 2024.

[4] Shunsuke Maeda, Yusuke Kaneko, Hideaki Muramatsu, Yukihiro Murakami, and Momoko Hayamizu. Orienting undirected phylogenetic networks to tree-child network, 2023. `https://arxiv.org/abs/2305.10162`.

[5] Laurent Bulteau, Mathias Weller, and Louxin Zhang. On turning a graph into a phylogenetic network. 2023.

[6] Gabriel Cardona, Francesc Rossello, and Gabriel Valiente. Comparison of Tree-Child Phylogenetic Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(4):552–569, 2009.

A practical FPT algorithm for the problem of orienting undirected phylogenetic networks

Tsuyoshi Urata[1], Manato Yokoyama[1], Momoko Hayamizu[2]
[1]Department of Pure and Applied Mathematics, Graduate School of Fundamental Science and Engineering, Waseda University, [2]Department of Applied Mathematics, Faculty of Science and Engineering, Waseda University
e-mail : uratsuyo244@moegi.waseda.jp

1 Abstract

Given an undirected phylogenetic network N, the $\mathcal{C}$-Orientation problem asks whether it is possible to orient N to a directed phylogenetic network $\vec{N}$ of a desired network class $\mathcal{C}$. In this talk, based on [1], we describe an FPT algorithm for $\mathcal{C}$-Orientation and a fast heuristic that can be used for Tree-Child Orientation. We analyse the time complexity of the proposed methods and compare the empirical performance with the state-of-the-art algorithm.

2 Notation and terminology

We recall the necessary definitions from [1]. An *undirected binary phylogenetic network* on X is a simple, connected, undirected graph N such that its vertex set V is partitioned into $V_I := \{v \in V \mid deg_N(v) = 3\}$ and $V_L := \{v \in V \mid deg_N(v) = 1\}$, and V_L can be identified with X. Each vertex in V_I and in V_L is called an *internal vertex* and a *leaf* of N, respectively.

A *directed binary phylogenetic network* on X is a simple, acyclic directed graph D such that the underlying graph of D is connected, the vertex set V of D contains a unique vertex ρ with $(indeg_D(\rho), outdeg_D(\rho)) = (0, 2)$ and the set $V \setminus \{\rho\}$ is partitioned into $V_T := \{v \in V \mid (indeg_D(v), outdeg_D(v)) = (1, 2)\}$ and $V_R := \{v \in V \mid (indeg_D(v), outdeg_D(v)) = (2, 1)\}$, and $V_L := \{v \in V \mid (indeg_D(v), outdeg_D(v)) = (1, 0)\}$ that can be identified with X. The vertex ρ is called *the root* of D, and each vertex in V_T, in V_R and in V_L is called a *tree vertex*, a *reticulation* and a *leaf* of D, respectively. A directed binary phylogenetic network D on X is a *tree-child network* if every non-leaf vertex of D has at least one child that is either a tree vertex or a leaf.

3 Orientation problems

Recently, Huber *et al.* [2] defined and discussed the following two problems.

Problem 1. *(*Degree-Constrained Orientation*)*

INPUT: *An undirected phylogenetic network $N = (V, E)$ on X, an edge $e_\rho \in E$ into which a unique root ρ is inserted, and the desired in-degree $\delta_N^-(v)$ of each $v \in V$.*

OUTPUT: *A directed phylogenetic network $\vec{N}$ that is an orientation of N and satisfies the constraint (e_ρ, δ_N^-) if it exists, and 'NO' otherwise.*

Problem 2. *($\mathcal{C}$-Orientation)*

INPUT: *An undirected binary phylogenetic network $N = (V, E)$ on X.*

OUTPUT: *An orientation $\vec{N}$ of N such that $\vec{N}$ belongs to the class $\mathcal{C}$ of directed binary phylogenetic networks on X if it exists, and 'NO' otherwise.*

Huber *et al.* [2] provided a $O(|E|)$ time algorithm for solving Problem 1. Regarding Problem 2, they described a simple exponential time algorithm and FPT algorithms for a special case of the problem. However, it remains challenging to develop a practical method for Problem 2. This motivates us to explore a heuristic approach.

4 Proposed methods for $\mathcal{C}$-Orientation

Let $N = (V, E)$ be an instance of Problem 2. Then, any orientation $\vec{N}$ of N contains $r = |E| - |V| + 1$ reticulations, and so we need to decide which r vertices to be reticulations in $\vec{N}$. The number of ways to select r vertices from the vertex set V is $\binom{|V|}{r}$. Theorem 1 allows us to reduce the search space from $\binom{|V|}{r}$ to $\prod_{i=1}^{r} |V(C_i)|$, where C_i are cycles in a basic cycle $\mathcal{S}$ of N.

Theorem 1. *Let (N, e_ρ, δ_N^-) be an instance of Problem 1 where N is binary and let V_R denote the set of reticulations specified by $\delta_N^-(v)$. If there exists an orientation $\vec{N}$ of N satisfying the constraint (e_ρ, δ_N^-), then for any cycle basis $\mathcal{S}$ of N, there exists a bijection $\phi : \mathcal{S} \to V_R$ with the property that $\phi(C) \in V(C)$ holds for each $C \in \mathcal{S}$.*

Based on Theorem 1, we propose an exact FPT algorithm for $\mathcal{C}$-Orientation (Problem 2). Our algorithm first computes a minimal cycle basis $\mathcal{S} = \{C_1, \ldots, C_r\}$ of N. It then iteratively selects one reticulation vertex from each basic cycle to form V_R, and solves Problem 1 for each (e_ρ, V_R) until a $\mathcal{C}$-orientation is found or all possibilities are exhausted. While this approach is still exponential, it significantly reduces the search space compared to previous methods, making it more practical for many instances. This algorithm is FPT both in the reticulation number r and the size c of longest basic cycles in $\mathcal{S}$ used in the computation.

For orientation to tree-child networks, we also propose a heuristic method that places reticulations as far apart as possible. Instead of computing $\prod_{i=1}^{r} |V(C_i)|$ reticulation placements, it maximizes the sum of distances between reticulations, avoiding exhaustive search. While not guaranteed to be correct for all instances, this approach works correctly for small r.

Acknowledgements We thank Yufeng Wu and Louxin Zhang for sharing code for randomly generating graphs, which we adapted for our experiments. We also appreciate Haruki Miyaji's helpful discussions.

参考文献

[1] T. Urata, M. Yokoyama and M. Hayamizu. Orientability of undirected phylogenetic networks to a desired class: Practical algorithms and application to tree-child orientation, arXiv:2407.09776 (2024).

[2] K. T. Huber, L. van Iersel, R. Janssen, M. Jones, V. Moulton, Y. Murakami and C. Semple. Orienting undirected phylogenetic networks. J. Comput. Syst. Sci. 140 (2024):103480.

距離行列から系統ネットワークを作る NJ-like ヒューリスティックスの開発

横山 慎人 [1], 菊地 祐太 [1], 早水 桃子 [2]
[1] 早稲田大学 基幹理工学研究科 数学応用数理専攻
[2] 早稲田大学 理工学術院
e-mail : mana.aki.aya@akane.waseda.jp

1　概要

Neighbor-Joining (NJ) 法 [1] は入力距離行列を実現する系統樹を推定する方法であり, 生物のデータから進化の道筋を可視化する方法として知られている.

また, NJ 法を拡張し, より広いクラスである系統ネットワークを出力する方法として, Neighbor-Net (NN) 法 [2] が知られている. NN 法により, NJ 法では発見できなかった進化の道筋の可能性を確認することが可能となった.

一方で, NN 法には, 出力されるグラフが複雑になることで, 進化の解釈が難しくなるという課題がある. 本講演では, NJ 法を改変することで, NN 法よりも解釈しやすい系統ネットワークを生成する新たなヒューリスティックスを提案し, 出力されるグラフの複雑さを NJ 法, NN 法と比較する.

2　Neighbor-Joining 法

Neighbor-Joining 法は, X 上の距離行列 D を入力とし, D を実現する系統樹 T を出力する距離行列法である. 星形系統樹を初期状態とし, 各ステップで最小の合計枝長となるペアを選択する. x_i と x_j をペアとして括り出したときの合計枝長 S_{ij} は,

$$S_{ij} = \frac{1}{2(n-1)} \sum_{k=3}^{n} (D(x_i, x_k) + D(x_j, x_k)) + \frac{1}{2} D(x_i, x_j) + \frac{1}{n-2} \sum_{2<i<j} D(x_i, x_j) \tag{1}$$

で求められる. 選択されたペアに対して新たな共通祖先を追加し, 距離行列を更新する. 距離行列の更新には, x_i, x_j でない任意の x_m と, x_i, x_j の共通祖先 x_{ij} との距離を $D_{m,ij}$ とし, x_i, x_j と x_{ij} のそれぞれの距離を $D_{i,ij}$, $D_{j,ij}$ として,

$$D_{m,ij} = \frac{1}{2} (D(x_i, x_m) + D(x_j, x_m) - D(x_i, x_j)) \tag{2}$$

$$D_{i,ij} = \frac{1}{2} (D(x_i, x_j) + D(x_i, Z) - D(x_j, x_Z)) \tag{3}$$

$$D_{j,ij} = \frac{1}{2} (D(x_i, x_j) + D(x_j, Z) - D(x_i, x_Z)) \tag{4}$$

を用いる. ただし, ここで Z は, $D(x_i, Z) = \frac{1}{n-2} \sum_{j=3}^{n} (x_i, x_j)$.

3　Neighbor-Net 法

Neighbor-Net 法は, X 上の距離行列 D を入力とし, D を実現する系統ネットワークを出力する距離行列法である. NN 法は次の 2 つのステップで進行する.

1) 近隣の頂点ペアを結合する操作を行い, X の circular ordering を決定する.
2) 各 circular split の重みを決定する計算を行う.

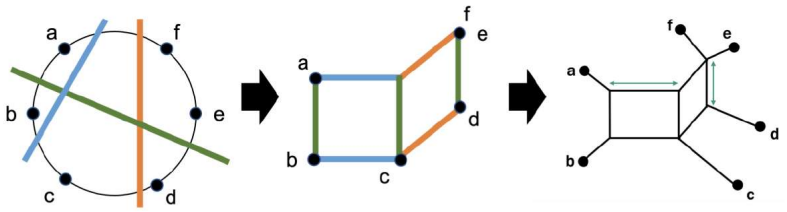

図 1. NN 法の各ステップのイメージ図. circular ordering により新たな進化の道筋の発見が可能となるが, サイクル数が増加し, 複雑なグラフとなる.

4　提案手法

提案手法では,「任意の四点距離空間の最適な実現は, 長方形とペンダントで記述でき, グラフの重みは一意に定まる [3] 」に注目した. 選ばれた各ペアに対応する長方形に着目して, 保存するか木とみなすかを決定する.

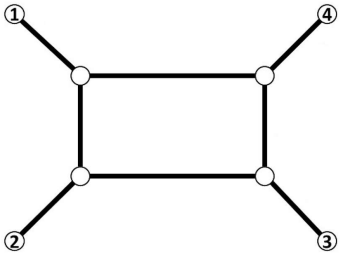

図 2. 選ばれた各ペア {1, 2} に対する長方形とペンダントの図. 残りの 2 点は新たなペア {3, 4} を選ぶ.

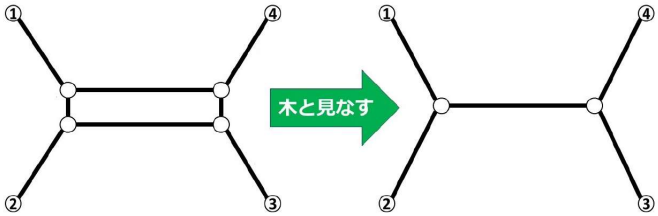

図 3. 長方形を木とみなす際のイメージ図.

謝辞　本研究にあたり, NJ 法の実装に協力してくださった浦田剛志さんに感謝いたします。また、議論に参加してくださった早水研究室のメンバーの方々にも重ねて感謝いたします。

参考文献

[1] N. Saitou and M. Nei, The Neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees, Molecular biology and evolution, Vol. 4, No. 4, (1987), 406–425.

[2] David Bryant and Vincent Moulton, Neighbor-Net: An Agglomerative Method for the Construction of Phylogenetic Networks, Molecular biology and evolution, Vol. 21, No. 2, (2004), 255–265.

[3] Bandelt, Hans-Jürgen and Dress, Andreas WM, A canonical decomposition theory for metrics on a finite set, dvances in mathematics, Vol. 92, No. 4, (1992), 47–105

