<table>
<thead>
<tr>
<th>タイトル</th>
<th>進化生物学における離散最適化問題の解法について 祖先形質復元問題に対する線形時間アルゴリズム 計算モデルと計算の複雑さに関する研究</th>
</tr>
</thead>
<tbody>
<tr>
<td>オリジナリスト</td>
<td>成嶋 弘</td>
</tr>
<tr>
<td>サイドネーム</td>
<td>数理解析研究所講究録</td>
</tr>
<tr>
<td>タイトル</td>
<td>メインタイトル</td>
</tr>
<tr>
<td>サイドネーム</td>
<td>メインサイドネーム</td>
</tr>
<tr>
<td>タイトル</td>
<td>メインタイトル</td>
</tr>
<tr>
<td>サイドネーム</td>
<td>メインサイドネーム</td>
</tr>
<tr>
<td>タイトル</td>
<td>メインタイトル</td>
</tr>
<tr>
<td>サイドネーム</td>
<td>メインサイドネーム</td>
</tr>
<tr>
<td>タイトル</td>
<td>メインタイトル</td>
</tr>
<tr>
<td>サイドネーム</td>
<td>メインサイドネーム</td>
</tr>
</tbody>
</table>
A Short History for MPR Problems

I) 解法について

J. S. Farris

･･･→ 1970 : Syst. Zoology

D. L. Swofford - W. P. Maddison

･･･→ 1987 : Math. Biosci. ([2])

M. Hanazawa - H. Narushima - N. Minaka

･･･→ 1991.8 : A Problem introduced by N. Minaka

･･･→ 1992.10 : Fifth Franco - Japanese Days

on Combinatorics and Optimization

･･･→ 1994.2 : 冬の LA symposium (京大数解研究研究集会)

･･･→ 1995 : Discrete Applied Math. ([4])

M. Hanazawa - H. Narushima

･･･→ 1994.6 : 7th Franco - Japanese Days

on Combinatorics, Optimization and Com. Geo. ([5])

･･･→ 1996.1 :

冬の LA symposium (京大数解研究研究集会) (This paper and [6])

II) ACTRAN, DELTRAN, and MPR-poset について

D. L. Swofford - W. P. Maddison

･･･→ 1987 : Math. Biosci. ([2])

N. Minaka

･･･→ 1993 : Natural History Research (自然誌研究) ([7])
N. Misheva - H. Narushima

\cdots \rightarrow 1994.3: 数学会年会応用数学分科会 ([9])
\cdots \rightarrow 1995.9: 第 7 回 RAMP シンポジウム ([10], [11])

III) 関連論文

多数 (現在、調査整理中) あり。

進化生物学と分類学について

馬渡俊輔 編著, 『動物の自然史』, 北海道大学図書刊行会 (1995)
特に, 三中信宏 著
第 4 章「分岐分析にもとづく系統推定の論理とその応用
— 系統樹推定と祖先形質復元 —」
Wagner parsimony criterion
= The criterion of maximum parsimony
Most Parsimonious Reconstruction (最節約復元)

MPR 問題

The problems are as follows: for a given el-tree \( T \)

1. determine \( L^*(T) \),
2. find any one MPR on \( T \),
3. enumerate all MPRs on \( T \),
4. obtain the MPR-sets for all internal nodes in \( T \),
5. problems on the ACCTRAN reconstruction on \( (T_s, r) \),
6. problems on the DELTRAN reconstruction on \( (T_s, r) \),
7. problems on the MPR-poset \( (\text{Rmp}(T), \leq) \),
8. etc \ldots.

アルゴリズムについて

Ha-Na-Mi algorithm ([4]) から
The key Lemma 1, Lemma 2 and Theorem 1 ([6]) を経て
Ha-Na algorithm ([6]) へ。
The Result for the computational complexities

<table>
<thead>
<tr>
<th>Problem No</th>
<th>Complexity Order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ha-Na-Mi</td>
</tr>
<tr>
<td>1</td>
<td>$n$</td>
</tr>
<tr>
<td>2</td>
<td>$n$</td>
</tr>
<tr>
<td>3</td>
<td>exp.</td>
</tr>
<tr>
<td>4</td>
<td>$n^2$</td>
</tr>
</tbody>
</table>

$n$: The number of nodes

The key points

1. Two pass algorithm
   - The first pass: Bottom-up
   - The second pass: Top-down
2. The $i$-th smallest number selection
   - The median $2\ l_2$ points of $2n$ numbers
   - The median $4\ l_2$ points of $2n$ numbers

Notations

1. $\Omega := \text{the set } \mathbb{R} \text{ of real numbers}
   \quad \text{or the set } \mathbb{N} \text{ of nonnegative integers}
2. an $\lambda$-tree $T = (V : V_O \cup V_H, E, \sigma)$
   \quad $\sigma : V_O \rightarrow \Omega$
   \quad $V_O := \text{the set of leaves}$
   \quad $V_H := \text{the set of internal nodes}$
3. a reconstruction on $T := \text{an assignment } \lambda : V \rightarrow \Omega \ (\lambda|_{V_O} = \sigma)$
4. $T|_{\lambda} := \text{an } \lambda\text{-tree } T \text{ under the reconstruction } \lambda$
5. the length $l(e)$ of a branch $e = \{u, v\}$ in $T|\lambda := |\lambda(u) - \lambda(v)|$
6. the length $L(T|\lambda) := \sum_{e \in E} l(e)$
7. $L^*(T) := \min\{L(T|\lambda) \mid \lambda \text{ is a reconstruction on } T\}$
8. an MPR (Most Parsimonious Reconstruction) on an $\lambda$-tree $T$
    \quad := \text{a reconstruction } \lambda \ (L(T|\lambda) = L^*(T))$
9. $\Rmp(T) := \text{the set of all MPRs on an } \lambda\text{-tree } T$
10. The MPR-set $S_u$ of a node $u := \{\lambda(u) \mid \lambda \in \Rmp(T)\}$
1 Introduction

For over a century, biologists have attempted to infer the evolutionary trees whose leaves are present day species. When constructing a tree, points of interest are the topology, when the transformation occurred, the length of the branches as well as the length of the tree itself. In the last four decades, the mathematical and algorithmic aspects of tree construction have been investigated.

Recent development of the theory of phylogeny enables scientists to estimate more precisely the evolutionary history of organisms. One of the main points of research is the reconstruction of ancestral character states on a given phylogeny under the criterion of maximum parsimony. This is known as character state optimization. Which basically means that character states are assigned to the internal nodes of a phylogenetic tree so as to minimize the total amount of evolutionary change, that is the length of the tree.

In phylogenetic analysis the optimization problem of assigning character states to the hypothetical ancestors of an evolutionary tree under the principle of maximum parsimony is known as the Most Parsimonious Reconstruction problem (MPR-problem). The biological and cladistic implications of this problem are beyond the scope of this paper. Rather we examine the mathematically formulated problem from a combinatorial point of view. In general, the MPR-problem is discussed under a given possible transformation relation of character states. For the MPR-problem under a rather general transformation relation, there is the dynamic programming method, i.e., a generalized algorithm which is essentially a brute-force method examining all possible assignments, which is described in [3]. In the paper [4], the MPR-problem and the related problems under linearly ordered states are discussed and an efficient method for this case is presented. In this paper, we present a more efficient algorithm for one of MPR problems than that in [4].

We use the (slightly refined) notations in Hanazawa-Narushima-Minaka [4]. We use $\Omega$ to denote the set that may be either the set $\mathbb{R}$ of real numbers or the set $\mathbb{N}$ of nonnegative integers. Let $T = (V = V_O \cup V_H, E, \sigma)$ be any tree with the leaves evaluated by a weight function $\sigma : V_O \rightarrow \Omega$, where $V$ is the set of nodes, $V_O$ is the set of leaves, $V_H$ is the set of internal nodes, and $E$ is the set of branches. We call this tree an el-tree, where "el" is an abbreviation of "evaluated leaf". For an el-tree $T$, we define an assignment $\lambda : V \rightarrow \Omega$ such that $\lambda|V_O$ (the restriction of $\lambda$ to $V_O$) $= \sigma$, where $\lambda(v)$ is called a state of $v$ under $\lambda$. This assignment is called a reconstruction on an el-tree $T$. For each branch $e$ in $E$ of an el-tree $T$ with a reconstruction $\lambda$, we define the length $l(e)$ of branch $e = \{u, v\}$ by $|\lambda(u) - \lambda(v)|$. Then the length $L(T|\lambda)$ of an el-tree $T$ under the reconstruction $\lambda$ is the sum of the lengths of the branches. That is $L(T|\lambda) = \sum_{e \in E} l(e)$. Furthermore we define the minimum length
$L^*(T)$ of $T$ by

$$L^*(T) = \min \{ L(T|\lambda) \mid \lambda \text{ is a reconstruction on } T \}.$$ 

Note that $L^*(T)$ is well-defined. A **Most Parsimonious Reconstruction** denoted by MPR on an el-tree $T$ is a reconstruction $\lambda$ such that $L(T|\lambda) = L^*(T)$. Generally an el-tree $T$ has more than one MPR. The set $\{ \lambda(u) \mid \lambda \text{ is an MPR on } T \}$ of states is called the **MPR-set** of a node $u$ and written as $S_u$.

The problems are as follows: for a given el-tree $T$

1. determine $L^*(T)$,
2. find any one MPR on $T$,
3. enumerate all MPRs on $T$,
4. obtain the MPR-sets for all internal nodes in $T$.

These problems are called the MPR problems in [4]. For their meanings in phylogeny, the reader may refer to Swofford-Maddison [2]. J. S. Farris, D. L. Swofford and W. P. Maddison have succeeded in solving the case of completely bifurcating trees. Hanazawa-Narushima-Minaka [4] present a solution for the MPR problems by introducing the concept of median interval obtained from sorting the endpoints of closed intervals, and then, discuss the computational complexity of their algorithms. In this paper, we present a more efficient algorithm for the problem 4. Compared with the previous algorithm in [4], the new algorithm has two main improved points. One is related to computing the median interval in the second pass of the algorithm. The other is in obtaining the MPR-sets, that is, the complexity of the previous algorithm in [4] for Problem 4 is $O(n^2)$ for the number $n$ of nodes in a given el-tree, but that of the new algorithm is $O(n)$.

## 2 The Key Lemmas and The Theorem

We denote the set $\{1, 2, \ldots, n\}$ of $n$ elements by $[n]$. Let $a_i$ ($i \in [2n]$) be any elements in $\Omega$, and be sorted in ascending order as follows:

$$x_1 \leq x_2 \leq \cdots \leq x_n \leq x_{n+1} \leq \cdots \leq x_{2n}.$$ 

Then we call $x_n$ and $x_{n+1}$ the **median two points** of the numbers $a_i$ ($i \in [2n]$), and denote $\langle x_n, x_{n+1} \rangle$ by

$$\text{med}2(a_1, a_2, \ldots, a_{2n}) \text{ or } \text{med}2(a_i : i \in [2n]).$$

We also call $x_{n-1}, x_n, x_{n+1}$ and $x_{n+2}$ the **median four points** of the numbers $a_i$ ($i \in [2n]$), and denote $\langle x_{n-1}, x_n, x_{n+1}, x_{n+2} \rangle$ by

$$\text{med}4(a_1, a_2, \ldots, a_{2n}) \text{ or } \text{med}4(a_i : i \in [2n]).$$
Lemma 1 Let $a$ and $b_i$ ($i \in [2m]$) be any elements in $\Omega$. Then
\[
\text{med}2(a, a, b_i : i \in [2m]) = \text{med}2(a, a, \text{med}4(b_i : i \in [2m])).
\]

Proof. Let $\text{med}4(b_i : i \in [2m]) = \langle x_{m-1}, x_m, x_{m+1}, x_{m+2} \rangle$. One may examine all possible cases with respect to $a$, $x_{m-1}$, $x_m$, $x_{m+1}$ and $x_{m+2}$. If $a \leq x_{m-1}$, then we have
\[
\text{the left side} = [x_{m-1}, x_m] = \text{the right side}.
\]
One can easily check the other four cases in a similar way.

Lemma 2 Let $a, b$ and $c_i$ ($i \in [2m]$) be any elements in $\Omega$. If $a \leq b$, then
\[
\min(\text{med}2(a, a, c_i : i \in [2m])) \leq \min(\text{med}2(b, b, c_i : i \in [2m]))
\]
and
\[
\max(\text{med}2(a, a, c_i : i \in [2m])) \leq \max(\text{med}2(b, b, c_i : i \in [2m])).
\]

Proof. Let $\text{med}4(c_i : i \in [2m]) = \langle x_{m-1}, x_m, x_{m+1}, x_{m+2} \rangle$. Then from lemma 1, we see that it is sufficient to examine all possible cases with respect to $a$, $b$, $x_{m-1}$, $x_m$, $x_{m+1}$ and $x_{m+2}$. If $a \leq x_{m-1}$ and $b \leq x_{m-1}$, then we have
\[
\min(\text{med}2(a, a, c_i : i \in [2m])) = z_{m-1} \leq z_{m-1} = \min(\text{med}2(b, b, c_i : i \in [2m]))
\]
and
\[
\max(\text{med}2(a, a, c_i : i \in [2m])) = z_m \leq z_m = \max(\text{med}2(b, b, c_i : i \in [2m])).
\]
One can easily check the other cases in a similar way.

Let $I_i = [a_i, b_i]$ ($i \in [m]$) be any family of closed intervals in $\Omega$. Let the median two points of all the endpoints $a_i$ and $b_i$ of $I_i$ ($i \in [m]$) be $x_m$ and $x_{m+1}$, i.e.,
\[
\text{med}2(a_i : i \in [m], b_i : i \in [m]) = \langle x_m, x_{m+1} \rangle.
\]
Then we call the closed interval $[x_m, x_{m+1}]$ in $\Omega$ the median interval of the closed intervals $I_i$ ($i \in [m]$), which is the key concept in a series of our papers, and denote it by
\[
\text{med}\{I_1, I_2, \ldots, I_m\} \text{ or } \text{med}\{I_i : i \in [m]\}.
\]

Let $T = (V, E)$ be a rooted (directed) tree, where $V$ is the set of nodes and $E (\subseteq V \times V)$ is the set of branches. For each $u$ and $v$ in $V$, we write $u \rightarrow v$ or $u = p(v)$ when $(u, v) \in E$, that is, when $u$ is a parent of $v$ (or $v$ is a child of $u$). For each $u$ and $v$ in $V$, $u$ is called an ancestor of $v$ (or $v$ is called a descendant of $u$), written $u \Rightarrow v$, if there is a sequence
of nodes \( u = u_1, u_2, \ldots, u_n = v \) in \( V \) such that \( u_i \rightarrow u_{i+1} (i \in [n-1]) \), which is called a \textit{path} in \( T \). Note that the relation "\( \Rightarrow \)" on \( V \) with the additional relation \( u \Rightarrow u \) for each \( u \) in \( V \) (the reflexive law) is a partial ordering on \( V \) and the relation "\( \rightarrow \)" results in a so-called covering relation on \( V \). We call a leaf (a node without a child) of a rooted tree a \textit{sink} to avoid ambiguity. For each \( u \) in \( V \), we denote a \textit{subtree} of \( T \) induced from a subset \( \{ v \in V | u \Rightarrow v \} \) of \( V \) by \( T_u = (V_u, E_u) \). Note that \( u \) is the root of \( T_u \).

Let \( T = (V_O \cup V_H, E, \sigma) \) be an el-tree rooted at \( r \) in \( V = V_O \cup V_H \). In addition, if \( r \) is a leaf, i.e., \( r \in V_O \) and \( s \) is its unique child, we denote the rooted tree by \( (T_s, r) \) to visualize the structure. In this case, the subtree \( T_s \) is called the \textit{body} of the tree \( T \); otherwise, i.e., if the root is not a leaf, the body of \( T \) is \( T \) itself.

For each node \( u \) in the body of a rooted el-tree \( T \), we assign a closed interval \( I(u) \) of \( \Omega \) recursively as follows:

\[
I(u) = \begin{cases} 
[\sigma(u), \sigma(u)] & \text{if } u \text{ is a sink}, \\
\text{med}(I(v) : u \rightarrow v) & \text{otherwise}.
\end{cases}
\]

We call \( I(u) \) the \textit{characteristic interval} of a node \( u \) and so \( I \) is called the \textit{characteristic interval map} on \( T \).

From the results of Theorem 1 in [4], we see that \( \text{med}([\lambda(p(u)), \lambda(p(u))], I(v) : u \rightarrow v) \) is the MPR-set of node \( u \) under the restriction that \( \lambda(p(u)) \) has been assigned to \( u \)'s parent \( p(u) \). We denote this subset of \( S_u \) by \( S_u|\lambda(p(u)) \). That is

\[
S_u|\lambda(p(u)) = \text{med}([\lambda(p(u)), \lambda(p(u))], I(v) : u \rightarrow v).
\]

Since it is easily determined, we often use it in our discussion and it figures in many of our results.

**Theorem 1** Let \( u \) be any internal node of a rooted el-tree \( T \) and the MPR-set \( S_u \) of \( u \) be \([a, b]\). Then the MPR-set \( S_v \) of \( v \) such that \( u \rightarrow v \) is

\[
S_v = [\min(S_v|a), \max(S_v|b)].
\]

**Proof.** First of all, from Corollary 5 in [4] we see that each MPR-set is a closed interval in \( \Omega \). From Theorem 3(ii) in [4] we also see that

\[
\bigcup_{x \in S_u} S_v|x = S_v.
\]

Then by Lemma 1 the family \( \{S_v|x | x \in S_u\} \) of closed intervals results in a snake chain. Therefore we have

\[
\min(S_v) = \min(S_v|a) \text{ and } \max(S_v) = \max(S_v|b) \quad \Box
\]
3 Computational Complexities

We now state the results on computational complexity of our algorithms. The number of comparisons required to "select" the \( i \)-th smallest of \( n \) numbers is essential in the complexity analysis of our algorithms. Therefore, the time complexity analysis is based on the following result for the selection algorithm called PICK by Blum et al [1].

**PICK Theorem.** The number \( f(i, n) \) of comparisons required to select the \( i \)-th smallest of \( n \) numbers is at most a linear function of \( n \), i.e., \( f(i, n) = O(n) \). □

Let's recall the definition of Minimum Length Map \( l^* \) and the proof of Theorem 4 in the previous paper [4]. Then we see that Lemma 1 does not have any effect on the algorithm (in [4]) for Problem 1. Let's recall the algorithm (in [4]) for Problem 2 and 3. The algorithm is a two-pass algorithm which consists of the first pass (bottom-up): the determination of the characteristic interval map \( I \) on a rooted el-tree \( T \) defined recursively in the direction from the sinks to the root and the second pass (top-down): the determination of each MPR on \( (T) \) defined recursively in the direction from the root to sinks. As the first pass has been already reviewed, we here review the second pass. Let \( T \) be a rooted el-tree \((T_s, r)\). Let \( I \) be the characteristic interval map on \( T \), which is already defined in the first pass. Let \( \lambda_{<u>} \) denote the restriction \( \lambda|V_u \) of a reconstruction \( \lambda \) on \( T \) to a subtree \( T_u \) of \( T \). Then the set \( \operatorname{RMP}2(r, s) \) of all most parsimonious reconstructions on \( T \) is defined recursively as follows: \( \lambda_{<s>} \in \operatorname{RMP}2(r, s) \) if and only if (1) \( \lambda(s) \in \operatorname{med}([\lambda(r), \lambda(r)], I(t) : s \rightarrow t) = S_s|\lambda(r) \) and (2) for each \( t \) such that \( s \rightarrow t \), \( \lambda_{<t>} \in \operatorname{RMP}2(s, t) \). Note that \( \lambda_{<s>} \) (with \( \lambda(r) = \sigma(r) \)) can be considered a reconstruction on \( T \).

The essential part in both of the two passes is the computation of median intervals. Speaking more concretely, the key part of the first pass is the computing of the median two points for defining \( \operatorname{med}(I(v) : u \rightarrow v) = I(u) \) for any internal node \( u \) under \( I(v)(u \rightarrow v) \) already defined, and that of the second pass is the computing of the median two points for defining \( \operatorname{med}([\lambda(u), \lambda(u)], I(w) : v \rightarrow w) = S_v|\lambda(u) \) for each child \( v \) of any interval node \( u \) under \( \lambda(u) \) and \( I(w) (v \rightarrow w) \) already defined. We now have two cases of making use of Lemma 1 and making no use of Lemma 1 for computing the median two points in the second pass. The complexity analysis of the algorithm for Problem 2 in the case of making no use of Lemma 1 is already done in the previous paper [4], and the result is stated as Theorem 5 in [4]. Does anything happen to the complexity analysis for the case of making use of Lemma 1? We next describe briefly about it. By using Lemma 1, we get

\[
S_v|\lambda(u) = \operatorname{med}([\lambda(u), \lambda(u)], I(w) : v \rightarrow w)
= \operatorname{med}([\lambda(u), \lambda(u)], \operatorname{med}4(I(w) : v \rightarrow w)),
\]

where \( \operatorname{med}4(I(w) : v \rightarrow w) \) denotes the median four points of all endpoints of \( I(w)(v \rightarrow w) \).
and so the last expression denotes the median interval for the six numbers of $\lambda(u), \lambda(u)$ and the median four points. From this fact, we see that the computing of the second pass is simplified under the assumption that the median four points is already computed in the first pass. Therefore, if the computing of not the median two points but the median four points is done in the first pass, then in the case of making use of Lemma 1, the computational complexity of the first pass increases but that of the second pass decreases, comparing with each complexity in the case of making no use of Lemma 1. From the facts described above, we may conclude that each complexity of the two cases is clearly of same order, that is, $O(n)$ for the number $n$ of nodes in a given el-tree $T$, and that "which of the two cases improves on the coefficients more" depends on a given el-tree $T$. We leave a work of more precise analysis on the constants.

The second pass of the previous algorithm in [4] for Problem 4 is not a little complicated and it complexity is $O(n^2)$ for the number $n$ of nodes in a given el-tree, which is described in Theorem 6 in [4], but the second pass of the new algorithm based on Theorem 1 is much simplified. We now state the main theorem on the computational complexity in this paper.

**Theorem 2** The complexity of our algorithm based on Theorem 1 for Problem 4 is $O(n)$ for the number $n$ of nodes in a given el-tree.

**Proof.** We prove the case of making no use of Lemma 1 for $T = (T_s, r)$. The algorithm also consists of the two passes: the first pass (bottom-up) is to determine the characteristic interval map $I$ on $T$ defined recursively, and the second pass (top-down) is to determine the MPR-sets $S_v$ for all internal node $v$ in $T$. Let $\text{Comp}(A)$ denote the (time) complexity for computing a formula $A$. We know already that $\text{Comp}$(The first pass) is $O(n)$. So, we consider the complexity of the second pass. Under the assumption that for any internal node $v$ and its parent $u$, and for each child $w$ of $v$, the MPR-set $S_v = [a, b]$ and $I(w)$ are already defined, from the definition of $S_v|a$ and PICK Theorem we have

$$
\text{Comp}(\min(S_v|a)) = \text{Comp}(\min(\text{med}([a, a], I(w) : v \to w)))
$$

$$= f(m_v + 1, 2(m_v + 1)) \leq c_v m_v,
$$

where $m_v$ is the number of children of $v$ and $c_v$ is a sufficiently large constant. Also we can similarly evaluate $\text{Comp}(\max(S_v|b))$. Then from Theorem 1, we have

$$
\text{Comp}(S_v) = 2f(m_v + 1, 2(m_v + 1)) \leq 2c_v m_v.
$$

Therefore, we get

$$
\text{Comp}$(The second pass)$ = \sum_{v \in V_H} \text{Comp}(S_v) \leq c \sum_{v \in V_H} m_v = c(n - 2),
$$

where $c = \max\{2c_v \mid v \in V_H\}$. Thus, the complexity of our algorithm for Problem 4 is $O(n)$ since both complexities of the two passes are $O(n)$. For the case of making use of Lemma 1 we get the same result by a similar discussion previously done for Problem 2.

□
参考文献


