Applied Mathematics Letters 25 (2012) 1148–1152

Contents lists available at SciVerse ScienceDirect

Applied Mathematics Letters

journal homepage: www.elsevier.com/locate/aml

'Bureaucratic' set systems, and their role in phylogenetics

David Bryant [a], Mike Steel [b,*]

[a] Department of Mathematics and Statistics, University of Otago, Dunedin, New Zealand

[b] Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand

ARTICLE INFO

Article history:
Received 9 June 2011
Received in revised form 7 November 2011
Accepted 10 February 2012

Keywords:
Cluster
Hierarchy
Tree
Algorithm
Phylogenetics

ABSTRACT

We say that a collection $\mathcal{C}$ of subsets of X is *bureaucratic* if every maximal hierarchy on X contained in $\mathcal{C}$ is also maximum. We characterize bureaucratic set systems and show how they arise in phylogenetics. This framework has several useful algorithmic consequences: we generalize some earlier results and derive a polynomial-time algorithm for a parsimony problem arising in phylogenetic networks.

1. Bureaucratic sets and their characterization

In this work we introduce and study a class of set systems that arise in various ways from trees, graphs and intervals. We are interested in this class because it can provide a setting in which certain hard optimization problems can be solved efficiently, and we provide a particular example of this for a parsimony problem on phylogenetic networks.

We first recall some standard phylogenetic terminology (for more details, the reader can consult [1]). Recall that a *hierarchy* $\mathcal{H}$ on a finite set X is a collection of sets with the property that the intersection of any two sets is either empty or equal to one of the two sets.

A hierarchy is *maximum* if $|\mathcal{H}| = 2|X| - 1$, which is the largest possible cardinality. In this case $\mathcal{H}$ corresponds to the set of clusters $c(T)$ of some rooted binary tree T with leaf set X (a *cluster* of T is the set of leaves that are separated from the root of the tree by any vertex). A maximum hierarchy necessarily contains $\{x\}$ for each $x \in X$, as well as X itself; we will refer to these $|X| + 1$ sets as the *trivial clusters* of X. More generally, any hierarchy containing all the trivial clusters corresponds to the clusters $c(T)$ of a rooted tree T with leaf set X (examples of these concepts are illustrated in Fig. 1(a), (b)). Note that a hierarchy $\mathcal{H}$ is maximum if and only if (i) $\mathcal{H}$ contains all the trivial clusters, and (ii) each set $C \in \mathcal{H}$ of size greater than 1 can be written as a disjoint union $C = A \sqcup B$, for two (disjoint) sets $A, B \in \mathcal{H}$.

We now introduce a new notion.

Definition. We say that a collection $\mathcal{C}$ of subsets of a finite set X is a *bureaucracy* if (i) $\mathcal{C} \neq \emptyset$ and $\emptyset \notin \mathcal{C}$, and (ii) every hierarchy $\mathcal{H} \subseteq \mathcal{C}$ can be extended to a maximum hierarchy $\mathcal{H}'$ such that $\mathcal{H} \subseteq \mathcal{H}' \subseteq \mathcal{C}$. In this case, we also say that $\mathcal{C}$ is *bureaucratic*.

Simple examples of bureaucracies include two extreme cases: the set of clusters of a binary tree, and the set $\mathcal{P}(X)$ of all non-empty subsets of X. Notice that $\{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, b, c\}\}$ and $\{\{a\}, \{b\}, \{c\}, \{b, c\}, \{a, b, c\}\}$ are both bureaucratic subsets of $\mathcal{P}(X)$ for $X = \{a, b, c\}$ but their intersection, $\{\{a\}, \{b\}, \{c\}, \{a, b, c\}\}$, is not. In particular, for an arbitrary subset Y of $\mathcal{P}(X)$ (e.g. $Y = \{\{a\}, \{b\}, \{c\}, \{a, b, c\}\}$), there may not be a unique minimal bureaucratic subset of $\mathcal{P}(X)$ containing Y.

* Corresponding author.
E-mail addresses: david.bryant@otago.ac.nz (D. Bryant), mike.steel@canterbury.ac.nz, mathmomike@gmail.com (M. Steel).

0893-9659/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.aml.2012.02.026

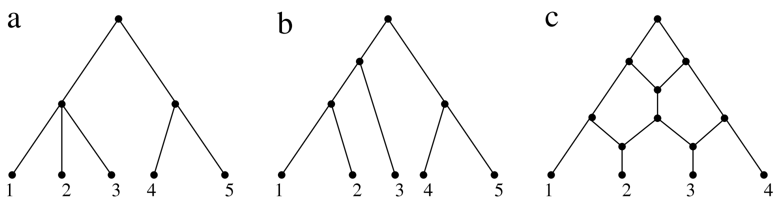

Fig. 1. (a) A rooted tree T with leaf set $X = \{1, 2, 3, 4, 5\}$, and with the cluster set $c(T)$ being equal to the hierarchy $\mathcal{H}$ consisting of the sets $\{1, 2, 3\}$, $\{4, 5\}$ and the trivial clusters. (b) A binary tree T with a cluster set consisting of $\mathcal{H} \cup \{\{1, 2\}\}$. (c) A binary and planar phylogenetic network $\mathcal{N}$ over $X = \{1, 2, 3, 4\}$ with a soft-wired cluster set $\mathrm{sw}(\mathcal{N})$ consisting of $\{1, 2\}$, $\{2, 3\}$, $\{3, 4\}$, $\{1, 2, 3\}$, $\{2, 3, 4\}$ and the trivial clusters.

In the next section we describe a more extensive list of examples, but first we describe some properties and provide a characterization of bureaucracies. In the following lemma, given two sets A and B from $\mathcal{C}$ we say that B *covers* A if $A \subsetneq B$ and there is no set $C \in \mathcal{C}$ with $A \subsetneq C \subsetneq B$.

Lemma 1. *If $\mathcal{C}$ is bureaucratic then:*

(i) *For any pair $A, B \in \mathcal{C}$, if B covers A then $B - A \in \mathcal{C}$.*
(ii) *For any $C \in \mathcal{C}$ with $|C| > 1$, we can write $C = A \sqcup B$ for (disjoint) sets $A, B \in \mathcal{C}$.*

Proof. For Part (i), suppose that $A, B \in \mathcal{C}$ and that B covers A. Let $\mathcal{H} = \{A, B\}$. Then $\mathcal{H}$ is a hierarchy that is contained within $\mathcal{C}$ and so there exists a maximum hierarchy $\mathcal{H}' \subseteq \mathcal{C}$ that contains $\mathcal{H}$. Note that A must be a maximal sub-cluster of B in $\mathcal{H}'$ (as otherwise B does not cover A) which requires that $B - A$ is a cluster of $\mathcal{H}'$ and thereby an element of $\mathcal{C}$.

For Part (ii), observe that the set $\mathcal{H} = \{C\}$ is a hierarchy, and the assumption that $\mathcal{C}$ is bureaucratic ensures the existence of a maximum hierarchy $\mathcal{H}' \subseteq \mathcal{C}$ containing $\mathcal{H}$, and so $\mathcal{H}'$ contains the required sets A, B. □

Note that the conditions described in Parts (i) and (ii) of Lemma 1, while they are necessary for $\mathcal{C}$ to be a bureaucracy, are not sufficient. For example, let $X = \{1, 2, 3, 4, 5, 6\}$ and let $\mathcal{C}$ be the union of

$$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{1, 2, 3\}, \{4, 5, 6\}, \{3, 4, 5\}, \{1, 2, 6\}, \{1, 5, 6\}, \{2, 3, 4\}\}$$

with the set of the seven trivial clusters. Then $\mathcal{C}$ satisfies Parts (i) and (ii) of Lemma 1, yet $\mathcal{C}$ is not bureaucratic since $\mathcal{H} = \{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$ does not extend to a maximum hierarchy on X using just elements from $\mathcal{C}$.

Theorem 2. *A collection $\mathcal{C}$ of subsets of X is bureaucratic if and only if it satisfies the following two properties:*

- (P1) *$\mathcal{C}$ contains all trivial clusters of X.*
- (P2) *If $\{C_1, C_2, \ldots, C_k\} \subseteq \mathcal{C}$ are disjoint and have union $\cup_i C_i$ in $\mathcal{C}$ then there are distinct i, j such that $C_i \cup C_j \in \mathcal{C}$.*

Proof. First suppose that $\mathcal{C}$ is bureaucratic. Then $\mathcal{C}$ contains a maximum hierarchy; in particular, it contains all the trivial clusters, and so (P1) holds. For (P2), suppose that $\mathcal{C}'$ is a collection of $k \geq 3$ disjoint subsets of X, each an element of $\mathcal{C}$, and $\bigcup \mathcal{C}' \in \mathcal{C}$. Then $\mathcal{H} = \mathcal{C}' \cup \{\bigcup \mathcal{C}'\}$ is a hierarchy. Let $\mathcal{H}' \subseteq \mathcal{C}$ be a maximum hierarchy on X that contains $\mathcal{H}$ (this exists, since $\mathcal{C}$ is bureaucratic) and let C be a minimal subset of X in $\mathcal{H}'$ that contains the union of at least two elements of $\mathcal{C}'$. Since $\mathcal{H}'$ is a maximum hierarchy, and $\bigcup \mathcal{C}' \in \mathcal{H}'$, C is precisely the union of exactly two elements of $\mathcal{C}'$; since $C \in \mathcal{H}' \subseteq \mathcal{C}$, this establishes (P2).

Conversely, suppose that a collection $\mathcal{C}$ of subsets of X satisfies (P1) and (P2), and that $\mathcal{H} \subseteq \mathcal{C}$ is a maximal hierarchy which is contained within $\mathcal{C}$. Suppose that $\mathcal{H}$ is not maximum (we will derive a contradiction). Then $\mathcal{H}$ contains a set C that is the disjoint union of $k \geq 3$ maximal proper subsets $A_1, \ldots, A_k$, each belonging to $\mathcal{H}$ (and thereby $\mathcal{C}$). Applying (P2) to $\mathcal{C}' = \{A_1, \ldots, A_k\}$, there exist two sets, say A_i, A_j for which $A_i \cup A_j \in \mathcal{C}$. So, if we let $\mathcal{H}' = \mathcal{H} \cup \{A_i \cup A_j\}$, then we obtain a larger hierarchy containing $\mathcal{H}$ that is still contained within $\mathcal{C}$, which is a contradiction. This completes the proof. □

2. Examples of bureaucracies

We have mentioned two extreme cases of bureaucracies, namely the set of clusters of a rooted binary tree having leaf set X, and the full power set $\mathcal{P}(X)$. Here are some further examples.

(1) The set of intervals of $[n] = \{1, 2, \ldots, n\}$ is a bureaucracy where an *interval* is a set $[i, j] = \{k : i \leq k \leq j\}$, $1 \leq i \leq j \leq n$.

Proof. Let $\mathcal{C}$ be the set of intervals of $[n]$. Then $\mathcal{C}$ contains the trivial clusters. Also, a disjoint collection $I_1, \ldots, I_k$, $k > 2$, of intervals has union an interval if and only if every element of $[n]$ between $\min \bigcup I_j$ and $\max \bigcup I_j$ lies in (exactly) one interval, in which case the union of any pair of consecutive intervals is an interval, so (P2) holds. By Theorem 2, $\mathcal{C}$ is bureaucratic. □

Similarly, if we order the elements of X in any fashion, we can define the set of *intervals on* X for that ordering by this construction (associating x_i with i), and can thus obtain a bureaucracy.

A natural question at this point is the following: Does the extension of intervals in a one-dimensional lattice (Example 1) to rectangles in a two-dimensional lattice also necessarily lead to bureaucracies? The answer is 'no' because condition (P2) can be violated due to the existence of subdivisions of integral sized rectangles into $k > 2$ disjoint squares of different integral sizes, the union of any two of which must therefore fail to be a rectangle (see e.g. [2]).

(2) Let T be a rooted tree (generally not binary) with leaf set X and let $\mathcal{C}$ be the set of all clusters compatible with all the clusters in $c(T)$. Then $\mathcal{C}$ is bureaucratic.

Proof. We have $\mathcal{C} = \{C \subseteq X : C \cap C' \in \{C, C', \emptyset\}$ for all $C' \in c(T)\}$.$\mathcal{C}$ is also the set of clusters that occur in at least one rooted phylogenetic tree on leaf set X that refines T (i.e. contains all the clusters of T), that is,

$$\mathcal{C} = \bigcup_{T' : c(T) \subseteq c(T')} c(T').$$

Suppose that $\mathcal{H} \subseteq \mathcal{C}$ is a hierarchy on X. Then $\mathcal{H} \cup c(T)$ is also a hierarchy on X since every element of $\mathcal{H}$ is compatible with every element of $c(T)$. Let $\mathcal{H}'$ be any maximum hierarchy on X containing $\mathcal{H}$. Then since $c(T) \subseteq \mathcal{H}'$, we have $\mathcal{H}' \subseteq \mathcal{C}$, and so, by definition, $\mathcal{C}$ is a bureaucracy. □

(3) Let $\mathcal{C}$ be a collection of subsets of X that includes the trivial clusters and which satisfies the condition

$$A, B \in \mathcal{C} \quad \text{and} \quad A \cap B \neq \emptyset \Rightarrow A \cup B \in \mathcal{C}. \tag{1}$$

Then $\mathcal{C}$ is bureaucratic if and only if $\mathcal{C}$ satisfies the covering condition in Lemma 1(i).

Before presenting the proof, we note that condition (1) is a weakening of the condition required for a 'patchwork' set system on X due to Andreas Dress and Sebastian Böcker (see e.g. [1], where the covering condition of Lemma 1(i) leads to an 'ample patchwork').

Proof. The 'only if' part follows from Lemma 1(i). Conversely, suppose that (1) holds for a set system $\mathcal{C}$ that includes all the trivial clusters of X and that satisfies the covering condition of Lemma 1(i). Suppose that $\mathcal{H} \subseteq \mathcal{C}$ is a maximal hierarchy contained within $\mathcal{C}$. We show that $\mathcal{H}$ is maximum. Suppose that this is not the case—we will derive a contradiction (by constructing a larger hierarchy $\mathcal{H}'$ containing $\mathcal{H}$ but still lying within $\mathcal{C}$). The assumption that $\mathcal{H}$ is not maximum implies that there exists a set $B \in \mathcal{H}$ which is the union of three or more disjoint sets $A_1, A_2, A_3, \ldots, A_k$, where $A_i \in \mathcal{H}$ (since the rooted tree associated with $\mathcal{H}$ has a vertex of degree $k \geq 3$). We consider two cases:
(i) B covers none of the sets from $A_1, A_2, A_3, \ldots, A_k$.
(ii) B covers one of the sets from $A_1, A_2, A_3, \ldots, A_k$.
We first show that Case (i) cannot arise under Condition (1). Suppose to the contrary that Case (i) arises. Then for $i = 1, \ldots, k$ there exists a set $C_i \in \mathcal{C}$ that contains A_i and which is covered by B. For any pair i, j with $i \neq j$, if $(B - C_i) \cap C_j = \emptyset$ then $C_j \subseteq C_i$. On the other hand, if $(B - C_i) \cap C_j \neq \emptyset$ then, by Condition (1), $(B - C_i) \cup C_j \in \mathcal{C}$, which means that $B = (B - C_i) \cup C_j$ (otherwise $(B - C_i) \cup C_j$ an element of $\mathcal{C}$ strictly containing C_j and strictly contained by B) and so $C_i \subseteq C_j$. Thus Case (i) requires that either $C_i \subseteq C_j$ or $C_j \subseteq C_i$, which implies (again by the assumption that B covers C_i and B covers C_j) that $C_i = C_j$. Since this identity holds for all distinct pairs i, j it follows that $C_1, C_2, \ldots, C_k$ are the same set C and this set contains $\bigcup_{i=1}^k A_i$ (since $A_i \subset C_i$). But then $B = \bigcup_{i=1}^k A_i \subseteq C$ which contradicts the assumption that B covers C_1 $(=C)$.

Thus only Case (ii) can arise. In this case, suppose that B covers A_i. By the assumption that $\mathcal{C}$ satisfies the covering condition described in Lemma 1(i), $B - A_i \in \mathcal{C}$ holds, and so we can take $\mathcal{H}' = \mathcal{H} \cup \{B - A_i\}$ which provides the required contradiction. □

(4) Let $G = (X, E)$ be a connected graph. Let $\mathcal{C}$ be the set of subsets $Y \subseteq X$ such that $G[Y]$ is connected (where $G[Y]$ is the subgraph formed by deleting vertices not in Y, together with their incident edges). Then $\mathcal{C}$ is bureaucratic.

Observe that taking G to be a linear graph recovers Example (1).

Proof. First note that $\mathcal{C}$ satisfies (P1), since G itself is connected, as is each vertex by itself. Now suppose that $A_1, \ldots, A_k$, $k > 2$, are disjoint clusters in $\mathcal{C}$ whose union, A, is also in $\mathcal{C}$. As $G[A]$ is connected, at least two clusters A_i, A_j must contain adjacent vertices, in which case $G[A_i \cup A_j]$ is connected and $A_i \cup A_j \in \mathcal{C}$. The result now follows by Theorem 2.

An alternative proof is to apply Example (3) and note that $\mathcal{C}$ satisfies Condition (1) and the covering condition of Lemma 1(i). □

(5) Let $\mathcal{C}$ be a *maximum weak hierarchy*, that is, a collection of non-empty subsets of X such that for all $A_1, A_2, A_3 \in \mathcal{C}$ the intersection $A_1 \cap A_2 \cap A_3$ equals at least one of $A_1 \cap A_2$, $A_1 \cap A_3$, $A_2 \cap A_3$, and with $|\mathcal{C}| = \binom{|X|+1}{2}$ [3]. Then $\mathcal{C}$ is bureaucratic.

Proof. We prove the result by induction on $|X|$. The result holds trivially for $|X| = 2$. Suppose it holds for $|X| < n$, and that $|X| = n$. Consider disjoint $C_0, \ldots, C_d \in \mathcal{C}$, $d \geq 2$, such that $C_0 \cup \cdots \cup C_k \in \mathcal{C}$. We will show that there are C_i, C_j such that $C_i \cup C_j \in \mathcal{C}$, and so $\mathcal{C}$ is bureaucratic by Theorem 2 (condition (P1) applies automatically for any maximum weak hierarchy [3]). By Proposition 1 of [3], there is an ordering $x_0, x_1, \ldots, x_{n-1}$ of X such that $\mathcal{C}' := \{A \in \mathcal{C} : x_0 \notin A\}$ is a maximum weak hierarchy on $X \setminus \{x_0\}$, $\{x_1, \ldots, x_k\} \in \mathcal{C}'$ for $k \geq 1$, and $\mathcal{C} = \mathcal{C}' \cup \left\{\{x_i : 0 \leq i \leq k\} : 0 \leq k < n\right\}$. If

$x_0 \notin C_0 \cup \cdots \cup C_k$ then the result holds by induction. Otherwise, suppose that $x_0 \in C_0$ and so $C_0 = \{x_0, x_1, \ldots, x_k\}$ for some k. Suppose that x_{k+1} lies in one of the sets $C_i,\ i > 0$, say C_1. If there is an ℓ such that $C_1 = \{x_{k+1}, x_{k+2}, \ldots, x_\ell\}$ we are done, since $C_0 \cup C_1 = \{x_0, x_1, \ldots, x_{k+1}, \ldots, x_\ell\} \in \mathcal{C}$. Otherwise there is an $\ell > k+1$ such that x_ℓ is an element of one of the sets $C_i,\ i > 0$, say C_1, but $x_{\ell-1} \notin C_1$. However, putting $A_1 = \{x_0, x_1, \ldots, x_{k+1}\}$, $A_2 = \{x_1, \ldots, x_\ell\}$ and $A_3 = C_1$ gives $A_1 \cap A_2 \cap A_3 \notin \{A_1 \cap A_2, A_1 \cap A_3, A_2 \cap A_3\}$, and so this second case cannot arise. □

3. Algorithmic applications

3.1. Maximum weight hierarchies

In general, the problem of finding the largest hierarchy contained within a set of clusters is NP-hard [4]. The problem becomes trivial in a bureaucratic collection since all maximal hierarchies are maximum. Less obvious, however, is the fact that the problem of finding a hierarchy with maximum *weight* can also be solved in polynomial time.

Theorem 3. *Let $\mathcal{C}$ be a bureaucratic collection of clusters on X and let $w : \mathcal{C} \longrightarrow \mathbb{R}$ be a weight function on $\mathcal{C}$. The problem of finding the hierarchy $\mathcal{H} \subseteq \mathcal{C}$ such that $w(\mathcal{H}) = \sum_{A \in \mathcal{H}} w(A)$ is maximized can be solved in polynomial time.*

Proof. If there are any clusters $A \in \mathcal{C}$ with negative weight $w(A)$ then set their weights to zero. It follows then that the weight of any maximum hierarchy $\mathcal{H} \subseteq \mathcal{C}$ equals the weight of the maximum weight hierarchy contained within $\mathcal{H}$. The 'Hunting for Trees' algorithm of [5] (which uses dynamic programming to construct, for every cluster in $A \in \mathcal{C}$, the maximum weight hierarchy with clusters in $\{B \in \mathcal{C} : B \subseteq A\}$) can now be used to recover the maximum hierarchy of maximum weight. □

3.2. Parsimony problems on networks

Consider a set $\mathcal{C}$ of clusters on X and let $f : X \to \mathcal{A}$ be a function that assigns each element $x \in X$ a state $f(x)$ in a finite set $\mathcal{A}$ (f is referred to in phylogenetics as a (discrete) *character*). Suppose we have a non-negative function δ on $\mathcal{A} \times \mathcal{A}$ where $\delta(a, b)$ assigns a penalty score for changing state a to b for each pair $a, b \in \mathcal{A}$ (the default option is to take $\delta(a, b) = 1$ for all $a \neq b$ and $\delta(a, a) = 0$ for all a).

Given any rooted X-tree T, with vertex set V and arc set E, let $l(f, T, \delta)$ denote the *parsimony score* of f on T relative to δ; that is,

$$l(f, T, \delta) = \min_{F:V \to \mathcal{A}, F|X = f} \left\{ \sum_{(u,v) \in E} \delta(F(u), F(v)) \right\}.$$

In words, $l(f, T, \delta)$ is the minimum sum of δ-penalty scores that are required in order to extend f to an assignment of states to all the vertices of T. This quantity can be calculated for a given T by well-known dynamic programming techniques (see e.g. [1]). Let $l(f, \mathcal{C}, \delta)$ (respectively, $l_{\text{bin}}(f, \mathcal{C})$) denote the minimal value of $l(f, T, \delta)$ among all trees T (respectively, all *binary* trees) that have their clusters in $\mathcal{C}$. Then we have the following general result.

Theorem 4. *Suppose that $\mathcal{C}$ is contained within a bureaucratic collection $\mathcal{C}'$ of subsets of X and $f : X \to \mathcal{A}$. There is an algorithm for computing $l(f, \mathcal{C}, \delta)$ with running time polynomial in $n = |X|$, $|\mathcal{A}|$ and $|\mathcal{C}'|$. Moreover, the algorithm can be extended to construct a rooted phylogenetic X-tree having all its clusters in $\mathcal{C}$ and with parsimony score equal to $l(f, \mathcal{C}, \delta)$ in polynomial time.*

Proof. For any subset Y of X, let

$$\delta_Y(a, b) = \begin{cases} \delta(a, b), & \text{if } Y \in \mathcal{C}; \\ 0, & \text{if } Y \notin \mathcal{C} \text{ and } a = b; \\ \infty, & \text{otherwise;} \end{cases}$$

and for any rooted phylogenetic X-tree T, let

$$l'(f, T, \delta) := \min_{F:V \to \mathcal{A}, F|X = f} \left\{ \sum_{(u,v) \in E} \delta_{c(v)}(F(u), F(v)) \right\},$$

where $c(v)$ is the cluster of T associated with v.

Let $l'(f, \mathcal{C}', \delta)$ (respectively, $l'_{\text{bin}}(f, \mathcal{C}', \delta)$) be the minimal value of $l'(f, T, \delta)$ over all trees (respectively, all binary trees) with clusters in $\mathcal{C}'$. By the definition of δ_Y, we have

$$l(f, \mathcal{C}, \delta) = l'(f, \mathcal{C}', \delta), \tag{2}$$

and by the assumption that $\mathcal{C}'$ is bureaucratic we have

$$l'(f, \mathcal{C}', \delta) = l'_{\text{bin}}(f, \mathcal{C}', \delta), \tag{3}$$

since $l'(f, T, \delta) \geq l'_{\text{bin}}(f, T', \delta)$ if T' is any binary tree that refines T. We now describe how $l'_{\text{bin}}(f, \mathcal{C}', \delta)$ can be efficiently calculated by dynamic programming.

For an element $a \in \mathcal{A}$ and $Y \in \mathcal{C}'$, let $L'(Y, a)$ be the minimum value of $l'(f|Y, T, \delta)$ across all binary trees T having leaf set Y and clusters in $\mathcal{C}'$, in which the root is assigned state a.

For $|Y| = 1$, say $Y = \{y\}$, we have

$$L'(Y, a) = \begin{cases} 0, & \text{if } f(y) = a; \\ \infty, & \text{otherwise} \end{cases}$$

and for $Y \in \mathcal{C}$, $|Y| > 1$, we have

$$L'(Y, a) = \min_{Y_1, Y_2 \in \mathcal{C}', a_1, a_2 \in \mathcal{A}} \left\{ L'(Y_1, a_1) + \delta_{Y_1}(a, a_1) + L'(Y_2, a_2) + \delta_{Y_2}(a, a_2) : Y_1 \sqcup Y_2 = Y \right\}. \tag{4}$$

Now,

$$l'_{\text{bin}}(f, \mathcal{C}', \delta) = \min_{a \in \mathcal{A}} L'(X, a).$$

Notice that when one evaluates $L'(X, a)$ using the above recursion (Eq. (4)), it is sufficient to compute $L'(Y, a)$ for just the sets $Y \in \mathcal{C}'$ rather than all subsets of X, by the definition of L'.

Thus, in view of Eqs. (2) and (3), one can compute $l(f, \mathcal{C}, \delta)$ in time polynomial in $n = |X|$, $|\mathcal{A}|$ and $|\mathcal{C}'|$. Moreover, by suitable book-keeping along the way, one can construct a rooted binary phylogenetic X-tree with clusters in $\mathcal{C}'$ and with a parsimony score equal to $l_{\text{bin}}(f, \mathcal{C}', \delta)$; by collapsing all edges of this tree that have a δ-score equal to 0 we obtain a rooted phylogenetic X-tree with clusters in $\mathcal{C}$ and with parsimony score equal to $l(f, \mathcal{C}, \delta)$. □

We note that this result has been described in the particular case where $\mathcal{C}$ is the bureaucracy described in Example (2) above, and where f maps to a set A with only two elements [6]. We provide a second application, to phylogenetic networks, based on Example (1) above, of intervals as bureaucratic set systems.

Let $\mathcal{N}$ be a rooted binary phylogenetic network on X. We say that $\mathcal{N}$ is *planar* if it can be drawn in the plane such that all the leaves and the root all lie on the outer face [7]. Let $\text{sw}(\mathcal{N})$ denote the set of 'soft-wired' clusters in $\mathcal{N}$ (the union of the cluster sets of all trees embedded in $\mathcal{N}$; see e.g. [8]). A simple example is shown in Fig. 1(c).

Corollary 5. *Suppose that $\mathcal{N}$ is a binary and planar phylogenetic network on X, and $f : X \to \mathcal{A}$. There is an algorithm for computing $l(f, \text{sw}(\mathcal{N}))$ with running time polynomial in n.*

Proof. Let $x_1, \ldots, x_n$ be the ordering of X given by their positions around the outer face in a planar embedding of $\mathcal{N}$, where x_1 and x_n come immediately after and before the root. Then any tree T embedded in $\mathcal{N}$ can be ordered such that the leaves are in order $x_1, \ldots, x_n$, implying that the clusters of T are all of the form $\{x_i, x_{i+1}, \ldots, x_j\}$ for some $1 \leq i \leq j \leq n$. It follows that the set $\text{sw}(\mathcal{N})$ is contained in the set of intervals of $X = \{x_1, \ldots, x_n\}$ (Example 1, above). The corollary now follows from Theorem 4. □

4. Concluding comments

While it is beyond the scope of this short note, it could be of interest to characterize *maximal* bureaucratic set systems. The following computational question also seems of interest:

Question. Is there an algorithm for deciding whether or not $\mathcal{C}$ is bureaucratic that runs in time polynomial in $|\mathcal{C}|$ and $|X|$?

Acknowledgments

We thank the NZ Marsden Fund and the Allan Wilson Centre for Molecular Ecology and Evolution for supporting this work. We also thank the two anonymous reviewers for some helpful suggestions.

References

[1] C. Semple, M. Steel, Phylogenetics, Oxford University Press, 2003.
[2] R.L. Brooks, C.A.B. Smith, A.H. Stone, W.T. Tutte, The dissection of rectangles into squares, Duke Math. J. 7 (1940) 312–340.
[3] H.-J. Bandelt, A. Dress, Weak hierarchies associated with similarity measures—an additive clustering technique, Bull. Math. Biol. 51 (1) (1989) 133–166.
[4] W.H.E. Day, D. Sankoff, Computational complexity of inferring phylogeny from compatibility, Syst. Zool. 35 (1986) 224–229.
[5] D. Bryant, Hunting for trees in binary character sets, J. Comput. Biol. 3 (2) (1996) 275–288.
[6] D. Huson, M. Steel, J. Whitfield, Reducing distortion in phylogenetic networks, in: P. Buecher, B.M.E. Moret (Eds.), Proceedings of WABI, Workshop on Algorithms in Bioinformatics, 2006, in: Lecture Notes in Bioinformatics, vol. 475, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 150–161.
[7] C. Scornavacca, F. Zickman, D. Huson, Tanglegrams for rooted phylogenetic trees and networks, Bioinformatics 27 (2011) i248–i256. ISMB 2011.
[8] D.H. Huson, R. Rupp, C. Scornavacca, Phylogenetic Networks: Concepts, Algorithms and Applications, Cambridge Univ. Press, 2011.