‘BUREAUCRATIC’ SET SYSTEMS, AND THEIR ROLE IN PHYLOGENETICS

DAVID BRYANT AND MIKE STEEL

Abstract. We say that a collection $C$ of subsets of $X$ is bureaucratic if every maximal hierarchy on $X$ contained in $C$ is also maximum. We characterise 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

We first recall some standard phylogenetic terminology (for more details, the reader can consult [6]). Recall that a hierarchy $H$ 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; we also assume that $X \in H$.

A hierarchy is maximum if $|H| = 2|X| - 1$, which is the largest possible cardinality. In this case $H$ 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 $H$ is maximum if and only if (i) $H$ contains all the trivial clusters, and (ii) each set $C \in H$ of size greater than 1 can be written as a disjoint union $C = A \sqcup B$, for two (disjoint) sets $A, B \in H$.

We now introduce a new notion.

1991 Mathematics Subject Classification. 05C05; 92D15.
Definition: We say that a collection \( C \) of subsets of a finite set \( X \) is a \textit{bureaucracy} if (i) \( C \neq \emptyset \) and \( \emptyset \not\in C \), and (ii) every hierarchy \( H \subseteq C \) can be extended to a maximum hierarchy \( H' \) such that \( H \subseteq H' \subseteq C \). In this case, we say that the collection is \textit{bureaucratic}.

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

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 \( C \) we say that \( B \) \textit{covers} \( A \) if \( A \subseteq B \) and there is no set \( C \in C \) with \( A \subsetneq C \subseteq B \).

Lemma 1. If \( C \) is bureaucratic then:

(i) For any pair \( A, B \in C \), if \( B \) covers \( A \) then \( B = A \in C \).

(ii) For any \( C \in C \) with \( |C| > 1 \), we can write \( C = A \sqcup B \) for (disjoint) sets \( A, B \in C \).

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

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

Note that the conditions described in Parts (i) and (ii) of Lemma 1 while they are necessary for \( C \) to be a bureaucracy, are not sufficient. For example, let \( X = \{1, 2, 3, 4, 5, 6\} \) and let \( C \) 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 \( C \) satisfies Parts (i) and (ii) of Lemma 1, yet \( C \) is not bureaucratic since \( H = \{\{1, 2\}, \{3, 4\}, \{5, 6\}\} \) does not extend to a maximum hierarchy on \( X \) using just elements from \( C \).

Theorem 2. A collection \( C \) of subsets of \( X \) is bureaucratic if and only if it satisfies the following two properties:

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

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

Conversely, suppose that a collection \( C \) of subsets of \( X \) satisfies (P1) and (P2), and that \( H \subseteq C \) is a maximal hierarchy which is contained within \( C \). Suppose that \( H \) is not maximum (we will derive a contradiction). Then \( H \) contains a set \( C \) that is the disjoint union of \( k \geq 3 \) maximal proper subsets \( A_1, \ldots, A_k \), each belonging to \( H \) (and thereby \( C \)). Applying (P2) to
We have mentioned two extreme cases of bureaucracies, namely the set of clusters of a binary 
X-tree 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 \((\text{P2})\) 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 
1-dimensional lattice (Example 1) to rectangles in a 2-dimensional lattice also necessarily 
lead to bureaucracies? The answer is ‘no’ because condition \((\text{P2})\) 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 \( \mathcal{C} = \{C \subseteq X : C \cap C' \in \{C, C', \emptyset\} \text{ for all } C' \in c(T)\} \). \( \mathcal{C} \) is also the set of 
clusters that occur in at least one rooted phylogenetic \( X \)-tree that refines \( T \), 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 maximal 
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:

\( A, B \in \mathcal{C} \text{ and } A \cap B \neq \emptyset \Rightarrow A \cup B \in \mathcal{C} \).

Then \( \mathcal{C} \) is bureaucratic if and only if \( \mathcal{C} \) satisfies the covering condition in Lemma [4(i)].

Condition [4(i)] is a weakening of the condition required for a ‘patchwork’ set system on 
\( X \) due to Andreas Dress and Sebastian Böcker (see e.g. [4], where the covering condition 
of Lemma [4(i)] leads to an ‘ample patchwork’).

Proof. The ‘only if’ part follows from Lemma [4(i)]. Conversely, suppose that [4(i)] holds 
for a set system \( \mathcal{C} \) that includes all the trivial clusters of \( X \) and that satisfies the covering 
condition of Lemma [4(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
be a function that assigns each element $x \in X$ a state $f(x)$ in a finite set $A$ (if is referred to in phylogenetics as a (discrete) character). Suppose we have a non-negative function $\delta$ on $A \times A$

(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 $A_i \cup A_j \in \mathcal{C}$.

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. The problem becomes trivial in a bureaucratic collection since all maximal hierarchies are maximum. Less obvious, however, is the fact that 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 $w : \mathcal{C} \rightarrow \mathbb{R}$ 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 [1] 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 $f : X \rightarrow A$ be a function that assigns each element $x \in X$ a state $f(x)$ in a finite set $A$ (is referred to in phylogenetics as a (discrete) character). Suppose we have a non-negative function $\delta$ on $A \times A$
where $\delta(a, b)$ assigns a penalty score for changing state $a$ to $b$ for each pair $a, b \in A$ (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 $\delta$; that is,

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

In words, $l(f, T, \delta)$ is the minimum sum of $\delta$-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. [6]). Let $l(f, C, \delta)$ (respectively, $l_{\text{bin}}(f, C)$) denote the minimal value of $l(f, T, \delta)$ among all trees $T$ (respectively, all binary trees) that have their clusters in $C$. Then we have the following general result.

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

**Proof.** For any subset $Y$ of $X$, let

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

and for any rooted phylogenetic $X$-tree $T$, let

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

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

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

(2) $l(f, C, \delta) = l'(f, C', \delta),$

and by the assumption that $C'$ is bureaucratic we have:

(3) $l'(f, C', \delta) = l_{\text{bin}}'(f, C', \delta).$

We now describe how $l_{\text{bin}}'(f, C', \delta)$ can be efficiently calculated by dynamic programming.

For an element $a \in A$ and $Y \in C'$, 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 $C'$, 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 $|Y| > 1$, we have

(4) $L'(Y, a) = \min_{Y_1, Y_2 \in C': a_1, a_2 \in 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 \cup Y_2 = Y \right\}. $
Now, 
\[ l_{\text{bin}}(f, C', \delta) = \min_{a \in A} L'(X, a). \]

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

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

We note that this result has been described in the particular case when \( C \) is the bureaucracy described in example (2) above, and where \( f \) maps to a set \( A \) with only two elements [5]. We provide a second application, to phylogenetic networks, based on Example (1) above, of intervals as bureaucratic set systems.

Let \( N \) be a rooted binary phylogenetic network on \( X \). We say that \( N \) is planar if it can be drawn in the plane so that all the leaves and the root all lie on the outer face. Let \( sw(N) \) denote the set of ‘soft-wired’ clusters in \( N \) (the union of the cluster sets of all trees embedded in \( N \); see e.g. [4]). A simple example is shown in Fig. 1(c).

**Corollary 5.** Suppose that \( N \) is a binary and planar phylogenetic network on \( X \), and \( f : X \rightarrow A \). There is an algorithm for computing 
\( l(f, sw(N)) \) 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 \( N \), where \( x_1 \) and \( x_n \) come immediately after and before the root. Then any tree \( T \) embedded in \( N \) can be ordered so that the leaves are in order \( x_1, \ldots, x_n \), implying that the clusters of \( T \) can be ordered so 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 \( sw(N) \) is contained in the set of intervals of \( X = \{x_1, \ldots, x_n\} \) (Example 1, above). The corollary now follows from Theorem 4. \( \square \)

We end this paper by posing a computational problem.

**Question.** Is there an algorithm for deciding whether or not \( C \) is bureaucratic that runs in time polynomial in \( |C| \) and \( |X| \)?

**Acknowledgements:** We thank the NZ Marsden Fund and the Allan Wilson Centre for Molecular Ecology and Evolution for supporting this work.

**References**


David Bryant: Department of Mathematics and Statistics, University of Otago, Dunedin, New Zealand. Mike Steel: Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand

E-mail address: david.bryant@otago.ac.nz, mike.steel@canterbury.ac.nz