Computing an Evolutionary Ordering is Hard

Laurent Bulteau, Gustavo Sacomoto, Blerina Sinaimeri

Université Lyon 1; INRIA Rhône-Alpes; CNRS, UMR5558; Laboratoire de Biométrie et Biologie Evolutive

We study the problem of computing evolutionary orderings of families of sets, as introduced by Little and Campbell [1].

Definition 1. *Let $\mathcal{S}$ be a family of subsets of some universe U. We say that $\mathcal{S}$ is* evolutionary *if there exists an ordering of its sets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ such that:*

- *Each set brings a new element, i.e. $S_i \not\subseteq \bigcup_{j=1}^{i-1} S_j$*

- *Each set, except the first one, has an old element, i.e. $S_i \cap \bigcup_{j=1}^{i-1} S_j \neq \emptyset$*

The associated algorithmic problem is the following:

EVOLUTIONARY ORDERING
Input: A family of subsets $\mathcal{S}$ of some universe U.
Question: Is $\mathcal{S}$ evolutionary?

We determine the computational complexity of this problem.

Theorem 1. EVOLUTIONARY ORDERING *is NP-hard.*

By a reduction from 3-SAT. Consider a formula Φ with n variables and m clauses. Assume that each clause appears twice (i.e., Φ can be written $\Phi = \Phi' \wedge \Phi'$). This is not restrictive, 3-SAT is clearly hard even when restricted to this class of formulas. For ease of presentation, assume that each literal occurs exactly k times, and each clause has exactly 3 literals. Note that $2kn = 3m$.

The universe on which the sets are constructed contains the following $6n + 5m$ elements:

- $2n$ *assignment elements*, denoted x_i and $\bar{x}_i$ for each $1 \le i \le n$

- $2n + 1$ *trigger elements*, denoted t_i and $\bar{t}_i$ for each $1 \le i \le n$ and τ

- $2n$ *free elements*, denoted f_i and $\bar{f}_i$ for each $1 \le i \le n$

- $2kn$ *literal elements*, denoted ℓ_i^h and $\bar{\ell}_i^h$ for each $1 \le i \le n$ and $1 \le h \le k$

- $2m$ *clause elements*, denoted c_j and c_j' for each $1 \le j \le j'$

We now create the following sets:

- Two *triggering sets*:
$$T := \{\tau\}$$
$$T' := \{\tau, t_1, \bar{t}_1, t_2, \ldots, \bar{t}_n\}$$

- For each $1 \le i \le n$, define two *variable sets* and a *verification set*:
$$L_i := \{x_i, t_i, f_i, \ell_i^1, \ldots, \ell_i^k\}$$
$$\bar{L}_i := \{\bar{x}_i, \bar{t}_i, \bar{f}_i, \bar{\ell}_i^1, \ldots, \bar{\ell}_i^k\}$$
$$V_i := \{x_i, \bar{x}_i, c_1, c_1', c_2, c_2', \ldots c_m, c_m'\}$$

- For each $1 \le j \le m$, where the jth clause uses, say, literals ℓ_1^1, ℓ_2^1, $\bar{\ell}_3^1$, deifne two *clause sets*
$$C_j := \{\ell_1^1, \ell_2^1, \bar{\ell}_3^1, c_j\}$$
$$C_j' := \{c_j, c_j'\}$$

1

We prove that this collection of sets has an evolutionary ordering if, and only if, Φ is satisfiable.

If. Given a truth assignment, we simply give an ordering of the sets by *adding* them one by one.

- Start with the triggering sets T and T'. No condition need to be satisfied for $T = \{\tau\}$, and, in T', τ is old and t_1 is new.

- For each variable x_i, add L_i if x_i is assigned true, $\bar{L}_i$ otherwise. For each one, t_i (or $\bar{t}_i$) is old, and f_i (or $\bar{f}_i$) is new.

- For each clause c_j, add set C_j followed by C'_j. Since the clause is satisfied, some literal ℓ_i^h (or $\bar{\ell}_i^h$) must be assigned true, so the corresponding element in L_i (or $\bar{L}_i$) is old for set C_j. Element c_j is new for set C_j, and then old for set C'_j. Element c'_j is new for set C'_j.

- For each variable x_i, add the verification set V_i. Element c_1 is old. If x_i is assigned true (resp. false), then element $\bar{x}_i$ (resp. x_i) is new.

- For each variable x_i, add $\bar{L}_i$ if x_i is assigned true, L_i otherwise. For each one, t_i (or $\bar{t}_i$) is old, and f_i (or $\bar{f}_i$) is new.

Overall, we have an ordering of the sets where each one has an old and a new element: the set is evolutionary.

Only if. Assume that our family of sets is evolutionary and consider such an ordering. Note that $T = \{\tau\}$ must be the very first set of this ordering (since otherwise iot cannot contain both old and new elements). This means that all other sets have an old and a new element. Write $\mathcal{A}$ for the family of the sets L_i and $\bar{L}_i$ that appear before their corresponding verification sets V_i. We make the following observations.

First, for each clause c_j and each variable x_i, set C'_j appears before V_i. This is because $C'_j \subset V_i$.

Now, for each variable x_i, it is not possible to have both $L_i \in \mathcal{A}$ and $\bar{L}_i \in \mathcal{A}$. Otherwise, V_i would not have any new element, since $V_i \subseteq L_i \cup \bar{L}_i \cup \bigcup_{j=1}^{m} C'_j$ and each C'_j is already before V_i. Thus, we design a truth assignment such that x_i is true if $L_i \in \mathcal{A}$, and false otherwise. This way, for each L_i or $\bar{L}_i$ in $\mathcal{A}$, the corresponding literal (x_i or $\bar{x}_i$), is assigned true.

It remains to show that the assignment satisfies formula Φ. Consider each clause c_j. First, C_j appears before C'_j. Indeed, the only sets intersecting C'_j are C_j and each V_i. Remember that V_is appear after C'_j, so the old element in C'_j must be from C_j, and C'_j appears after C_j. It follows that the old element of C_j cannot be c_j, hence it is a literal element ℓ_i^h or $\bar{\ell}_i^h$. So the corresponding variable set L_i or $\bar{L}_i$ must be before C_j, hence before C'_j and V_i. Overall, for each clause, one of L_i or $\bar{L}_i$ corresponding to a literal of the clause is in $\mathcal{A}$, and the literal is satisfied by our assignment. So the whole formula Φ is satisfiable.

References

[1] Little, C. H. C., Campbell, A. E. Evolutionary Families of Sets. *The Electronic Journal of Combinatorics*, 7(R10), 2, 2000.