Computing an Evolutionary Ordering is Hard

Laurent Bulteau, Gustavo Sacomoto, Blerina Sinaimeri

Computing an Evolutionary Ordering is Hard [1]

Laurent Bulteau [a,2], Gustavo Sacomoto [a,3] Blerina Sinaimeri [a,4]

[a] *INRIA Grenoble Rhône-Alpes, France*
UMR CNRS 5558
LBBE, Université Lyon 1, France

Abstract

A family of sets is evolutionary if it is possible to order its elements such that each set except the first one has an element in the union of the previous sets and also an element not in that union. This definition is inspired by a conjecture of Naddef and Pulleyblank concerning ear decompositions of 1-extendable graphs. Here we consider the problem of determining whether a family of sets is evolutionary. We show that the problem is NP-complete even when every set in the family has at most 3 elements and each element appears at most a constant number of times. In contrast, for families of intervals of integers, we provide a polynomial time algorithm for the problem.

Keywords: evolutionary family, ear-decomposition, complexity.

[1] This work was supported by the European Research Council under the European Community's Seventh Framework Programme (FP7 / 2007-2013) / ERC grant agreement no. [247073]10 and by the European Union Framework Program 7 "BacHBerry", Project number FP7-613793.

[2] Email: laurent.bulteau@inria.fr
[3] Email: gustavo.sacomoto@inria.fr
[4] Email: blerina.sinaimeri@inria.fr

1 Introduction

A family of sets $\mathcal{S}$, is called *evolutionary* if there exists an ordering of the sets in $\mathcal{S}$ such that except for the first set, every set in the order contains both an element that belongs to the union of the previous sets (an *old* element) and an element that does not (a *new* one). The concept of evolutionary families was introduced in [2] motivated by a result of Carvalho, Lucchesi and Murty [1] on a question related to the ear-decomposition of 1-extendable graphs, posed by Naddef and Pulleyblank in [4].

Let G be a graph and G' a subgraph of G. A path P in G of odd length is called an *ear* of G' if $V(P) \cap V(G')$ consists of the two endpoints of P. An *n-ear* is a set of n vertex-disjoint ears. A graph G is 1-*extendable* if each one of its edges belongs to a perfect matching of G.

It is well-known that bipartite graphs that are 1-extendable can be constructed simply by adding one ear at a time to smaller graphs of the same type. This property does not hold in general for 1-extendable graphs. In order to guarantee an ear decomposition where each graph in the sequence is also 1-extendable we have to allow to add more ears simultaneously. In [3], Lovász and Plummer proved the so called *Two Ear Theorem*, where they show that in constructing 1-extendable graphs, adding at most two (disjoint) ears in each step will suffice to guarantee that the intermediate graphs in the construction are themselves 1-extendable. Naddef and Pulleyblank [4] asked about the smallest number of 2-ear additions necessary in such a decomposition. This question was answered in [1] and their results inspired Little and Campbell [2] to introduce the concept of evolutionary families.

Indeed, it was proved in [1] that the minimum number of 2-ear additions in a ear decomposition of a 1-extendable graph G is $dim\mathcal{C}(G) - dim\mathcal{A}(G)$, where $\mathcal{C}(G)$ and $\mathcal{A}(G)$ denote the cycle space and the alternating subspace of G, respectively. Little and Campbell [2] then observed that the minimum number of 2-ear additions is achieved for a basis of $\mathcal{A}(G)$ that, among other properties, is evolutionary. They also gave some sufficient conditions for a family to be evolutionary.

In this paper, we investigate the complexity of deciding whether a given family is evolutionary. More formally, we consider the following problem:

Evolutionary Ordering Problem
INSTANCE: A family $\mathcal{S}$ of sets.
DECIDE: Is $\mathcal{S}$ evolutionary?

This problem was posed by M. Steel as one of the "Penny Ante" prize

questions of the Annual New Zealand Phylogenetics Meeting in 2012 [5]. In this paper we fully answer this question by showing in Section 2 that the problem is NP-complete and remains so even when every set in the family has at most 3 elements and where each element appears a constant number of times. Observe that this result is tight regarding the set cardinalities as for the case when each set has at most two elements the problem can be easily solved in polynomial time as already observed in [2].

Finally, in Section 3 we show that for sets of intervals of integers the problem is solvable in polynomial time.

2 Evolutionary ordering is NP-complete

We show in this section that deciding whether a given family is evolutionary is NP-complete. Moreover, it remains so even when each of the sets in the family has cardinality at most 3. The reduction is obtained using a restricted version of the SAT problem defined as follows:

$(3,2)$**-SAT** [6]
INSTANCE: A Boolean formula F in conjunctive normal form where each clause has two or three literals and each literal appears in at most 2 clauses.
DECIDE: Is there some assignment of true and false value that will make the formula F true?

Theorem 2.1 Evolutionary Ordering *is* NP*-complete.*

Proof. It is clear that the Evolutionary Ordering problem is in NP. We show it is also NP-complete. Let Φ be an instance of (3,2)-SAT with n variables and m clauses. For ease of presentation, assume that each literal occurs exactly 2 times, and each clause has exactly 3 literals. Note that $4n = 3m$.

We construct an instance of Evolutionary Ordering problem by assigning a set to each variable, literal and clause. The main idea is that every evolutionary ordering of the sets would necessary imply an ordering where the sets corresponding to variables that are assigned true come first, then the sets of the corresponding literals, then all the clauses that contain at least one literal assigned true and hence that appeared before, and finally it must be possible to add the remaining sets corresponding to the variables that are assigned false. Note, that in order to have an assignment we have to guarantee that not both sets corresponding to $x_i, \bar{x}_i$ are taken before the clause sets. We proceed now to formally detail the construction.

The universe in which the sets are constructed contains the following

$8n + 2m + 1$ elements:

- 1 *trigger element*, denoted τ
- $2n$ *assignment elements*, denoted x_i and $\bar{x}_i$ for each $1 \leq i \leq n$
- $2n$ *free elements*, denoted f_i and $\bar{f}_i$ for each $1 \leq i \leq n$
- $4n$ *literal elements*, denoted ℓ_i^1, ℓ_i^2 and $\bar{\ell}_i^1$, $\bar{\ell}_i^2$ for each $1 \leq i \leq n$
- $2m$ *clause elements*, denoted c_j and c'_j for each $1 \leq j \leq j'$

We create the following sets:

- A *triggering set*: $T := \{\tau\}$
- For each $1 \leq i \leq n$, define two *variable sets* and a *verification set*:

$$L_i := \{x_i, \tau, f_i, \ell_i^1, \ell_i^2\} \quad \text{and} \quad \bar{L}_i := \{\bar{x}_i, \tau, \bar{f}_i, \bar{\ell}_i^1, \ell_i^2\}$$

$$V_i := \{x_i, \bar{x}_i, c_1, c'_1, c_2, c'_2, \ldots c_m, c'_m\}$$

- For each $1 \leq j \leq m$, where the jth clause uses, say, literals ℓ_1^1, ℓ_2^1, $\bar{\ell}_3^1$, define two *clause sets*

$$C_j := \{\ell_1^1, \ell_2^1, \bar{\ell}_3^1, c_j\} \quad \text{and} \quad C'_j := \{c_j, c'_j\}$$

Let $\mathcal{S}$ denote this family of sets. We prove that $\mathcal{S}$ has an evolutionary ordering if, and only if, Φ is satisfiable.

If. Given a truth assignment, we simply give an evolutionary ordering of the sets. Start with the triggering set $\{\tau\}$. No condition needs to be satisfied for this set.

For each variable x_i add the set L_i if x_i is assigned true, $\bar{L}_i$ otherwise. For each one of them, τ is old, and f_i (or $\bar{f}_i$) is new.

For each clause c_j, add sets C_j and 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 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 add the verification set V_i. For every i the elements c_i, c'_i are all old for each of those sets. If x_i is assigned true (resp. false), then element $\bar{x}_i$ (resp. x_i) is new.

Finally, we add the remaining variable sets. For each variable x_i add $\bar{L}_i$ if x_i is assigned true, L_i otherwise. For each one, τ 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. Consider an evolutionary ordering of the sets. Observe that every

such ordering would put set $T = \{\tau\}$ in the first position. Moreover, all the clauses C'_j must appear before any V_i as $C'_j \subset V_i$.

Let us write $\mathcal{A}$ for the family of the sets L_i and $\bar{L}_i$ that appear before their corresponding verification sets V_i. 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 regular 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 either C'_j is first or C_j is before 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. □

It is possible to modify the gadget used in the previous reduction in such a way that every set in the family has cardinality at most 3 and every element appears at most a constant number of times. Thus, the following stronger result holds.

Theorem 2.2 Evolutionary Ordering *is* NP*-complete even if each set in the family has cardinality at most* 3 *and each element appears at most a constant number of times.*

3 A polynomial algorithm for interval families

In this section we consider the particular case where the sets in $\mathcal{S}$ are intervals of integers, *i.e.* for each $S_i \in \mathcal{S}$ we have that $S_i = [a_i, b_i] = \{k \in \mathbb{Z} \mid a_i \leq k \leq b_i\}$. We obtain a polynomial algorithm for this case using a dynamic programing approach. Before describing the algorithm, we need some extra definitions. Given a family of intervals $\mathcal{S}$, we define $a(\mathcal{S}) = \min\{a_i \mid [a_i, b_i] \in \mathcal{S}\}$ and $b(\mathcal{S}) = \max\{b_i \mid [a_i, b_i] \in \mathcal{S}\}$. We say $S_i = [a_i, b_i]$ is the leftmost (resp. rightmost) interval if it is a unique interval with $a_i = a(\mathcal{S})$ (resp. $b_i = b(\mathcal{S})$). In other words, $a_j > a_i$ (resp. $b_j < b_i$) for all $j \neq i$.

An immediate property is that if the union of the sets in $\mathcal{S}$ is not an interval (precisely, the interval $[a(\mathcal{S}), b(\mathcal{S})]$), then $\mathcal{S}$ cannot be evolutionary. The key observation to obtain an efficient algorithm is given in the next lemma.

Lemma 3.1 *The last interval in any evolutionary ordering of $\mathcal{S}$ is a leftmost or rightmost interval. Moreover, if $\mathcal{S}$ has no leftmost nor rightmost intervals then $\mathcal{S}$ is not evolutionary.*

Proof. Suppose by contradiction that the last interval $[x, y]$ in an evolutionary order of $\mathcal{S}$ is not a leftmost or rightmost interval. Write $\mathcal{S}' = \mathcal{S} \setminus \{[x, y]\}$. This implies that $a(\mathcal{S}') = a(\mathcal{S})$ and $b(\mathcal{S}') = b(\mathcal{S})$. On the other hand, $[x, y]$ has a new element z. Thus, there is no evolutionary ordering for $\mathcal{S}'$, since this set contains intervals both to the left and to the right of z but none including it.□

As a corollary, we have the following recursive relation: $(\star)$ $\mathcal{S}$ is evolutionary iff $\mathcal{S}$ has a leftmost interval S_l and $\mathcal{S} \setminus \{S_l\}$ is evolutionary, or $\mathcal{S}$ has a rightmost interval S_r and $\mathcal{S} \setminus \{S_r\}$ is evolutionary.

Lemma 3.2 *Let $\mathcal{S}', \mathcal{S}'' \subseteq \mathcal{S}$ correspond to subproblems of $(\star)$, if $a(\mathcal{S}') = a(\mathcal{S}'')$ and $b(\mathcal{S}') = b(\mathcal{S}'')$ then $\mathcal{S}' = \mathcal{S}''$.*

Proof. At each step of $(\star)$ a leftmost (or rightmost) interval is removed from the current set $\mathcal{S}$, thus strictly increasing $a(\mathcal{S})$ (or decreasing $b(\mathcal{S})$). This implies that $\mathcal{S}'$ contains all intervals entirely included in $[a(\mathcal{S}'), b(\mathcal{S}')]$. Thus, $\mathcal{S}' = \mathcal{S}''$. □

It follows that the number of subproblems of $(\star)$ is bounded by the number of intervals of the form $[a_i, b_j]$, which in turn is bounded by $|\mathcal{S}|^2$. A standard dynamic programming approach based on $(\star)$ gives a polynomial algorithm.

References

[1] M. H. Carvalho, C. L. Lucchesi, and U. S. R. Murty, *Ear Decompositions of Matching Covered Graphs*, Combinatorica, 19 (2) (1999), pp. 151–174.

[2] C. H. C. Little and A. E. Campbell, *Evolutionary families of sets*, The Electronic Journal of Combinatorics, 7 (1) (2000).

[3] L. Lovász and M. D. Plummer, *Matching Theory*, Akadémiai Kiadó, Budapest (1986).

[4] D. J. Naddef and W. R. Pulleyblank *Ear Decompositions of Elementary Graphs and GF2-rank of Perfect Matchings*, North-Holland Mathematics Studies 66 (1982), pp. 241–260.

[5] M. Steel, *The "Penny Ante" 2012*, http://www.math.canterbury.ac.nz/bio/events/south2012/files//penny_ante_problems.pdf (2012).

[6] C. A. Tovey, *A simplified* NP*-complete satisfiability problem*, Discrete Applied Mathematics, 8 (1) (1984), pp. 85–89.