On Bubble Generators in Directed Graphs

Vicente Acuna, Roberto Grossi, Giuseppe Italiano, Leandro Lima, Romeo Rizzi, Gustavo Sacomoto, Marie-France Sagot, Blerina Sinaimeri

On Bubble Generators in Directed Graphs

V. Acuña[1], R. Grossi[2], G. F. Italiano[3], L. Lima[5], R. Rizzi[4], G. Sacomoto[5], M.-F. Sagot[5], and B. Sinaimeri[5]

[1] Center for Mathematical Modeling (UMI 2807 CNRS), University of Chile, Santiago, Chile.
viacuna@dim.uchile.cl
[2] Università di Pisa, Pisa, Italy and Erable, INRIA, France.
grossi@di.unipi.it
[3] Università di Roma "Tor Vergata", Roma, Italy.
giuseppe.italiano@uniroma2.it
[4] Università di Verona, Verona, Italy.
Romeo.Rizzi@univr.it
[5] INRIA Grenoble Rhône-Alpes, Université Lyon 1; CNRS, UMR5558, LBBE, Villeurbanne, France.
(leandro.ishi-soares-de-lima@inria.fr, gustavo.sacomoto@gmail.com, marie-france.sagot@inria.fr, blerina.sinaimeri@inria.fr)

Abstract. Bubbles are pairs of internally vertex-disjoint (s, t)-paths with applications in the processing of DNA and RNA data. For example, enumerating alternative splicing events in a reference-free context can be done by enumerating all bubbles in a de Bruijn graph built from RNA-seq reads [16]. However, listing and analysing all bubbles in a given graph is usually unfeasible in practice, due to the exponential number of bubbles present in real data graphs. In this paper, we propose a notion of a bubble generator set, *i.e.* a polynomial-sized subset of bubbles from which all the others can be obtained through the application of a specific symmetric difference operator. This set provides a compact representation of the bubble space of a graph, which can be useful in practice since some pertinent information about all the bubbles can be more conveniently extracted from this compact set. Furthermore, we provide a polynomial-time algorithm to decompose any bubble of a graph into the bubbles of such a generator in a tree-like fashion.

Keywords: Bubbles, Bubble generator set, Bubble space, Decomposition algorithm

1 Introduction

Bubbles are pairs of internally vertex-disjoint (s, t)-paths with applications in the processing of DNA and RNA data. For example, in the genomic context, genome assemblers usually identify and remove bubbles in order to remove sequencing errors and linearise the graph [14, 22, 18, 10]. However, bubbles can also represent interesting biological events, *e.g.* allelic differences (SNPs and indels) when

processing DNA data [7, 20, 21], and alternative splicing events in RNA data [17, 16, 15, 11]. Due to their practical relevance, several theoretical studies concerning bubbles were done in the past few years [1, 4, 13, 15, 19], usually related to bubble-enumeration algorithms, but the literature regarding this mathematical object remains small when compared to the literature on cycles, *i.e.* undirected eulerian subgraphs, which is a related concept.

In practice, due to the high throughput of modern sequencing machines, the genomic and transcriptomic de Bruijn graphs tend to be huge, usually containing from millions to billions of vertices. As expected, the number of bubbles also tends to be large, exponential in the worst case, and therefore algorithms that deal with them either simplify the graph by removing bubbles, or just analyse a small subset of the bubble space. Such subsets usually correspond to bubbles with some predefined characteristics, and may not be the best representative of the bubble space. More worrying is the fact that all the relevant events described by bubbles that do not satisfy the constraints are lost. On the other hand, any algorithm that tries to be more exhaustive, analysing a big part of the bubble space, will certainly spend a prohibitive amount of time in real data graphs and will not be applicable. This motivates further work for finding efficient ways to represent the information contained in the bubble space. In a graph-theoretical framework, one way to do this is to obtain a compact description of all bubbles.

In this paper, we propose a bubble generator, *i.e.* a "representative set" of the bubbles in a graph that allows to reconstruct all and only the bubbles in a graph. More specifically, we show how to identify, for any given directed graph G, a generator set of bubbles $\mathcal{G}(G)$ which is of polynomial size in the input, and such that any bubble in G can be obtained in a polynomial number of steps by properly combining the bubbles in the generator $\mathcal{G}(G)$ through some suitably defined graph operations. We also propose a polynomial-time decomposition algorithm that, given a bubble B in the graph G, finds a sequence of bubbles from the generator $\mathcal{G}(G)$ whose combination results in B. The latter algorithm can be applied when one needs to know how to decompose a bubble into its elementary parts, which are the bubbles in $\mathcal{G}(G)$, *e.g.* when identifying and decomposing complex alternative splicing events [17] into several elementary alternative splicing events.

This work was inspired by the studies on cycle bases, which represent a compact description of all the cycles in a graph. The study of cycle bases started a long time ago [12] and has attracted much attention in the last fifteen years, leading to many interesting results such as the classification of different types of cycle bases, the generalisation of these notions to weighted and to directed graphs, as well as several complexity results for constructing bases. We refer the interested reader to the books of Deo [5] and Bollobás [2], and to the survey of Kavitha *et al.* [8] for an in-depth coverage of cycle bases. However, it is worth mentioning some characteristics that make the problems related to bubble generators very different (and more difficult) from the ones related to cycle bases. Indeed, a cycle base in a directed graph contains cycles with orientations that can be arbitrary, so that elements in the base are not even directed cycles in the

original graph [9] (if the graph is strongly connected, then it is possible to find a cycle base composed only of directed cycles [6]). On the contrary, bubbles impose a particular orientation of the cycle. Observe that a cycle base composed solely of bubbles cannot be directly translated into a bubble generator, since such set represents the cycle space, which is a superset of the bubble space. In order to obtain a representative set of only the bubble space, it is required to change the symmetric difference operator, *i.e.* the operator used to combine two bubbles. The restriction we impose in this operator is that two bubbles are combinable if the output is also a bubble, *i.e.* the operator is undefined if the output is not a bubble. By imposing such restriction, the bubble space is not closed under the symmetric difference operator, and thus cannot be represented as a vector space over $\mathbb{Z}_2$, as is the case with the cycle space. As such, the algorithms developed for cycle bases in undirected and directed graphs do not apply to our problem with bubbles.

The remainder of the paper is organised as follows. Section 2 present some definitions that will be used throughout the paper. Section 3 introduces the bubble generator. Section 4 presents a polynomial-time algorithm for decomposing any bubble in a graph into elements of the generator set. Finally, we conclude with open problems in Section 5.

2 Preliminaries

Throughout the paper, we assume that the reader is familiar with the standard graph terminology, as contained for instance in [3]. A *directed* graph is a pair $G = (V, A)$, where V is the set of vertices, and A is the set of arcs. Given a graph G, we also denote by $V(G)$ the set of vertices of G, and by $A(G)$ the set of arcs of G. For convenience, we set $n = |V(G)|$ and $m = |A(G)|$. In this paper, all graphs considered are directed, unweighted, without parallel arcs and finite. An arc $a = (u, v)$ is said to be incident to vertices u and v. In particular, $a = (u, v)$ is said to be leaving vertex u and entering vertex v. Alternatively, $a = (u, v)$ is an outgoing arc for u and an incoming arc for v. The in-degree of a vertex v is given by the number of arcs entering v, while the out-degree of v is the number of arcs leaving v. The degree of v is the sum of its in-degree and out-degree.

We say that a graph $G' = (V', A')$ is a subgraph of a graph $G = (V, A)$ if $V' \subseteq V$ and $A' \subseteq A$. Given a subset of vertices $V' \subseteq V$, the subgraph of G induced by V', denoted by $G[V']$, has V' as vertex set and contains all arcs of G that have both endpoints in V'. Given a subset of arcs $A' \subseteq A$, the subgraph of G induced by A', denoted by $G[A']$, has A' as arc set and contains all vertices of G that are endpoints of arcs in A'. Given a subset of vertices $V' \subseteq V$ and a subset of arcs $A' \subseteq A$, we denote by $G - V'$ the graph $G[V \setminus V']$ and by $G - A'$ the graph $G[A \setminus A']$. Given two graphs G and H, their union $G \cup H$ is the graph F for which $V(F) = V(G) \cup V(H)$ and $A(F) = A(G) \cup A(H)$. Their intersection $G \cap H$ is the graph F for which $V(F) = V(G) \cap V(H)$ and $A(F) = A(G) \cap A(H)$.

Let s, t be any two vertices in G. A (*directed*) *path* from s to t in G is a sequence of vertices $s = v_1, v_2, \ldots, v_k = t$, such that $(v_i, v_{i+1}) \in A$ for

$i = 1, 2, \ldots, k-1$. We also allow a single vertex to be a path. A path is *simple* if it does not contain repeated vertices. A path from s to t is also referred to as an (s,t)-path. The length of a path p is the number of arcs in p and will be denoted by $|p|$. We write $p \subseteq q$ if p is a subpath of q. Given a path p_1 from x to y and a path p_2 from y to z, we denote by $p_1 \cdot p_2$ their concatenation, *i.e.* the path from x to z defined by the path p_1 followed by p_2. For a path $p = v_1, v_2, \ldots, v_k$, we say that the subpath $p_1 = v_1, \ldots, v_i$ ($p_2 = v_j, \ldots, v_k$) is a *prefix* (*suffix*) of p for some $1 \leq i \leq k$ ($1 \leq j \leq k$). Two paths $p = v_1, v_2, \ldots, v_k$ and $q = u_1, u_2, \ldots, u_l$ are vertex disjoint if they share no vertices. Further, if the subpaths $p_1 = v_2, \ldots, v_{k-1}$ of p and $q_1 = u_2, \ldots, u_{l-1}$ of q are vertex disjoint, we say that p and q are internally vertex disjoint. Throughout this paper, all the paths considered will be simple and referred to as paths.

Definition 1. *Given a directed graph G and two vertices $s, t \in V(G)$, not necessarily distinct, an* (s,t)-bubble *B consists of two (s,t)-paths that are internally vertex disjoint. Vertex s is the source and t is the target of the bubble. If $s = t$ then one of the paths of the bubble has length 0, and therefore B corresponds to a directed cycle. We then say that B is a* degenerate bubble.

In the following, we assume that shortest paths are unique. This is without loss of generality, and indeed there are many standard techniques for achieving this, including perturbing arc weights by infinitesimals. However, for our goal, it suffices to use a "lexicographic ordering". Namely, we define an arbitrary ordering $v_1, \ldots, v_n$ on the vertices of G. A path p is considered lexicographically shorter than a path q if the length of p is strictly smaller than the length of q, or, if p and q have the same length, the sequence of vertices associated to p is lexicographically smaller than the sequence associated to q. We denote this by $p <_{lex} q$.

We denote by $B = (p, q)$ the bubble having p, q as its two internally vertex-disjoint paths, referred to as *legs*. We denote by $\ell(B)$ (resp., by $\mathcal{L}(B)$) the shorter (resp., longer) between the two legs p, q of B. We also denote by $|B|$ the number of arcs of bubble B. Note that $|B| = |\ell(B)| + |\mathcal{L}(B)|$.

Next, we define a total order on the set of bubbles.

Definition 2. *Let B_1 and B_2 be any two bubbles. B_1 is* smaller *than B_2 (in symbols, $B_1 < B_2$) if one of the following holds: either (i) $\mathcal{L}(B_1) <_{lex} \mathcal{L}(B_2)$; or (ii) $\mathcal{L}(B_1) = \mathcal{L}(B_2)$ and $\ell(B_1) <_{lex} \ell(B_2)$.*

3 The bubble generator

As with cycle bases in undirected graphs, we define a symmetric difference operator, but which operands are bubbles. Given two bubbles B_1 and B_2 of a directed graph G, the constrained symmetric difference operator Δ is such that $B_1 \Delta B_2$ is defined if and only if $G[(A(B_1) \cup A(B_2)) \setminus (A(B_1) \cap A(B_2))]$ is a bubble. Otherwise, we say that $B_1 \Delta B_2$ is undefined. If $B_1 \Delta B_2$ is defined, we also say that B_1 and B_2 are *combinable*. Given two combinable bubbles B_1 and B_2, we refer

to $B_1 \Delta B_2$ as the *sum of* B_1 *and* B_2, and denote it also by $B_1 + B_2$. We also say that the bubble $B_1 + B_2$ is *generated* from bubbles B_1 and B_2, and that it can be *decomposed* into the bubbles B_1 and B_2.

Let G be a directed graph and let $\mathcal{B}$ be a set of bubbles in G. The set of all the bubbles that can be generated starting from bubbles in $\mathcal{B}$ is called the *span* of $\mathcal{B}$. A set of bubbles $\mathcal{B}$ is called a *generator* if each bubble in G is spanned by $\mathcal{B}$, *i.e.* it can be recursively decomposed down to bubbles of $\mathcal{B}$. Due to our constrained symmetric difference operator Δ, all subgraphs generated by the elements in $\mathcal{B}$ are necessarily bubbles. Since not all pairs of bubbles of G are combinable, the bubble space is not closed under Δ, and therefore it does not form a vector space over $\mathbb{Z}_2$.

Definition 3. *A bubble B is* composed *if it can be obtained as a sum of two smaller bubbles. Otherwise, the bubble B is called* simple.

For a directed graph G, we denote by $\mathcal{S}(G)$ the set of simple bubbles of G. It is not difficult to see that $\mathcal{S}(G)$ is a generator. For now, we are not able to: 1) prove that $\mathcal{S}(G)$ can be found in polynomial time or if it is $\mathcal{NP}$-Hard to do so; 2) prove that any bubble in G can be obtained in a polynomial number of steps from bubbles in $\mathcal{S}(G)$. Nevertheless, we introduce next another generator $\mathcal{G}(G) \supseteq \mathcal{S}(G)$ which can be found in polynomial time and for which we can prove that any bubble in G can be obtained in a polynomial number of steps from the bubbles in $\mathcal{G}(G)$. Let $p : s = x_0, x_1, \ldots, x_h = t$ be a path from s to t and let $0 \leq i \leq j \leq h$. To ease the notation, we denote by $p_{i,j}$ the subpath of p from x_i to x_j, and refer also to $p_{0,j}$ as $p_{s,j}$ and to $p_{i,h}$ as $p_{i,t}$. The next theorem provides some properties of simple bubbles.

Theorem 1. *Let B be a simple (s,t)-bubble in a directed graph G. The following holds:*

(1) $\ell(B)$ is the shortest path from s to t in G;
(2) Let $\mathcal{L}(B) = s, v_1, \ldots, v_r, t$. Then $s, v_1, \ldots, v_r$ is the shortest path from s to v_r in G.

Proof. Let B be a simple (s,t)-bubble: we show that both conditions (1) and (2) must hold.

We first consider condition (1). If B is degenerate, then it trivially satisfies condition (1). Therefore, assume that B is non-degenerate and, by contradiction, that $\ell(B)$ is not the shortest path from s to t. Let $p^* : s = x_0, x_1, \ldots, x_h = t$ be the shortest path from s to t in G. For $0 \leq i \leq j \leq h$, by subpath optimality, $p^*_{i,j}$ is the shortest path from x_i to x_j. Let k be the smallest index, $0 \leq k < h$, for which the arc (x_k, x_{k+1}) does not belong to either one of the legs of B. Such an index k must exist, as otherwise p^* would coincide with a leg of B. Furthermore, let l, $k < l \leq h$, be the smallest index greater than k for which $x_l \in V(B)$. Such a vertex x_l must also exist, since $x_h = t \in V(B)$. In other words, x_k is the first vertex of the bubble B where p^* departs from B and x_l, $l > k$, is the first vertex where the shortest path p^* intersects again the bubble B. By definition of x_k

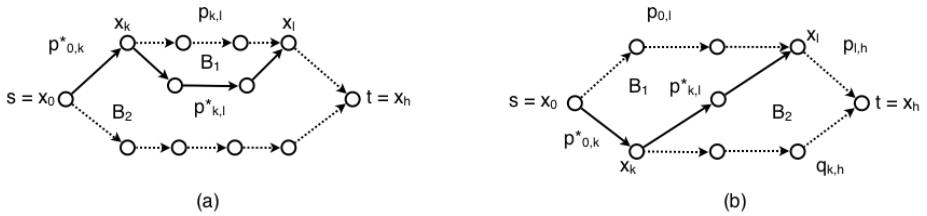

Fig. 1: Case (1) of the proof of Theorem 1. The prefix of the shortest path from s to t is shown as a solid line.

and x_l, $p^*_{k,l}$ is internally vertex-disjoint with both legs of B. We now claim that B can be obtained as the sum of two smaller bubbles, thus contradicting our assumption that B is a simple bubble.

To prove the claim, we distinguish two cases, depending on whether x_k and x_l are on the same leg of B or not. Consider first the case when x_k and x_l are on the same leg p of B (see Fig. 1(a)). Let B_1 be the bubble with $\ell(B_1) = p^*_{k,l}$ and $\mathcal{L}(B_1) = p_{k,l}$. First, note that if either $x_k \neq s$ or $x_l \neq t$, then $p_{k,l}$ is a proper subpath of a leg of B. Hence, $|\mathcal{L}(B_1)| = |p_{k,l}| < |\mathcal{L}(B)|$, and $B_1 < B$. Otherwise, suppose $s = x_k$ and $t = x_l$. Then either $\mathcal{L}(B_1) = \ell(B) <_{lex} \mathcal{L}(B)$, or $\mathcal{L}(B_1) = \mathcal{L}(B)$ and $\ell(B_1) = p^*_{k,l} = p^* <_{lex} \ell(B)$. In both cases, $B_1 < B$. Let B_2 be the bubble which is obtained from B by replacing $p_{k,l}$ by $p^*_{k,l}$ (see Fig. 1(a)). Since $p^*_{k,l}$ is the shortest path, by subpath optimality, $p^*_{k,l} <_{lex} p_{k,l}$, thus $B_2 < B$. As a result, B can be obtained as the sum of two smaller bubbles B_1, B_2, thus contradicting the assumption that B is simple.

Consider now the case where x_k and x_l are on different legs of B (see Fig. 1(b)). Notice that this means $x_k \neq s$ and $x_l \neq t$. Let p be the leg containing x_l and q the one containing x_k. Note that $p = p_{0,l} \cdot p_{l,h}$ and $q = p^*_{0,k} \cdot q_{k,h}$. Moreover, let B_1 be the bubble such that the two legs of B_1 are $p^*_{0,k} \cdot p^*_{k,l} <_{lex} q$ and $p_{0,l}$, which is a proper subpath of p. Hence, $B_1 < B$. Let B_2 be the bubble such that the two legs of B_2 are $q_{k,h}$, which is a proper subpath of q, and $p^*_{k,l} \cdot p_{l,h} <_{lex} p$. Hence, $B_2 < B$, and $B = B_1 + B_2$, which implies again that B is not simple.

We show now that B satisfies also condition (2). Assume, by contradiction, that B satisfies condition (1) but not (2), and so $p = s, v_1, \ldots, v_r$ (note that p is equal to $\mathcal{L}(B)$ without its last arc) is not the shortest path from s to v_r in G. Let $p^* : s = x_0, \ldots, x_{h-1} = v_r$, $p^* \neq p$, be such a shortest path in G. Similarly to the previous case, let k be the smallest index, $0 \leq k < h-1$, for which the arc (x_k, x_{k+1}) does not belong to either one of the legs of B, *i.e.* x_k is the first vertex where the shortest path p^* departs from B. Such an index k must exist, as otherwise p^* would coincide with a leg of B. Let $l, k < l \leq h-1$, be the smallest index such that $x_l \in V(B)$. Namely, x_l is the first vertex after x_k where the shortest path p^* intersects again bubble B. Such a vertex x_l must always exist, since $x_{h-1} = v_r \in V(B)$. Since $k < l$, we have that $|p^*_{k,l}| \geq 1$. Furthermore, we

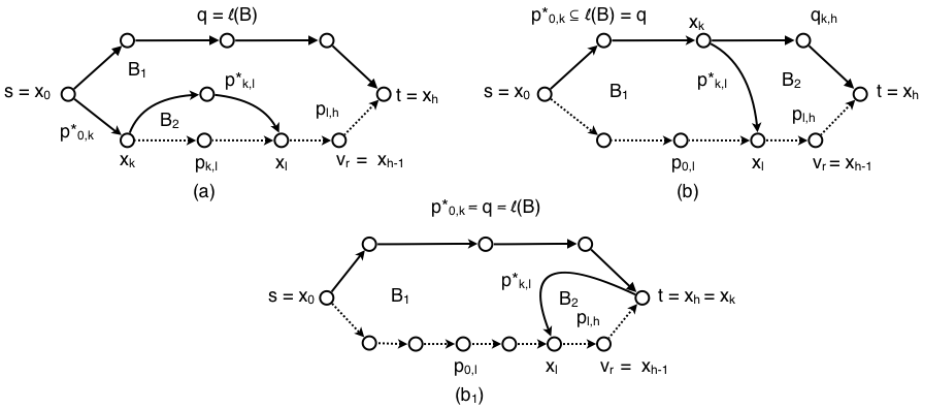

Fig. 2: Case (2) of the proof of Theorem 1. The shortest path from s to t and the prefix of the shortest path from s to v_r are shown as solid lines.

claim that x_l must be in $\mathcal{L}(B) - \{s, t\}$. If this were not the case, we would have two distinct shortest paths from s to x_l in G ($p^*_{0,l}$ and the subpath of $\ell(B)$ from $s = x_0$ to x_l), which contradicts our assumption that shortest paths are unique.

We again distinguish two cases: when both x_k, x_l belong to $\mathcal{L}(B)$, and when $x_k \in \ell(B)$ and $x_l \in \mathcal{L}(B)$. We set $p = \mathcal{L}(B), q = \ell(B)$.

In the first case (see Fig. 2(a)), let B_1 be the bubble with $\ell(B_1) = \ell(B)$ and $\mathcal{L}(B_1) = p^*_{0,k} \cdot p^*_{k,l} \cdot p_{l,h}$. Since $|p^*_{k,l}| <_{lex} |p_{k,l}|$ then $\mathcal{L}(B_1) <_{lex} \mathcal{L}(B)$, and thus $B_1 < B$. Let B_2 be the bubble with $\ell(B_2) = p^*_{k,l}$, and $\mathcal{L}(B_2) = p_{k,l}$. Since $\mathcal{L}(B_2) \subset \mathcal{L}(B)$ (as $x_k \neq t$), $B_2 < B$. As a result, B can be obtained as the sum of two smaller bubbles B_1, B_2, thus contradicting the assumption that B is simple.

In the second case (see Fig. 2(b)), let B_1 be the bubble with $\ell(B_1) = p^*_{0,k} \cdot p^*_{k,l}$ and $\mathcal{L}(B_1) = p_{0,l}$. Since $\mathcal{L}(B_1) \subset \mathcal{L}(B)$, $B_1 < B$. Let B_2 be the bubble with $\ell(B_2) = q_{k,h}$, and $\mathcal{L}(B_2) = p^*_{k,l} \cdot p_{l,h}$. Since $|\mathcal{L}(B_2)| < |\mathcal{L}(B)|$, $B_2 < B$. Again, B can be obtained as the sum of two smaller bubbles B_1, B_2, thus contradicting the assumption that B is simple. Finally, notice that this includes also the case $x_k = t$ and the argument holds identically with B_2 being a degenerate bubble. For the sake of clarity, we depicted this case separately in Fig. 2(b_1). ■

Given a directed graph G, we denote by $\mathcal{G}(G)$ the set of bubbles in G satisfying conditions (1) and (2) of Theorem 1.

Remark 1. Conditions (1) and (2) of Theorem 1 are not sufficient to guarantee that a bubble is simple, *e.g.* see Fig. 3. Thus, the generator $\mathcal{G}(G)$ is not necessarily minimal.

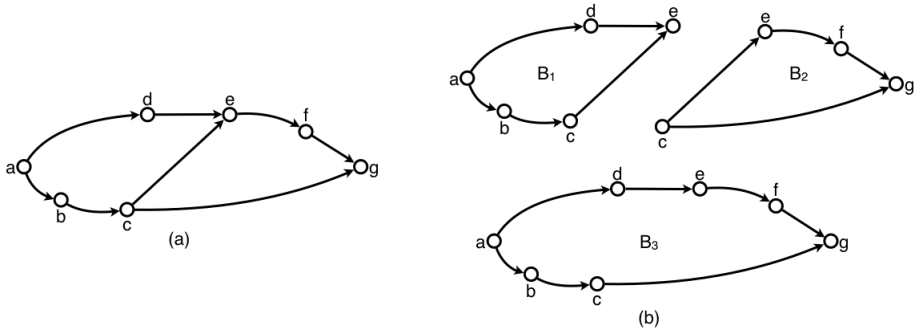

Fig. 3: An example showing that conditions (1) and (2) of Theorem 1 are not sufficient to guarantee that a bubble is simple. (a) A directed graph G. (b) The three bubbles B_1, B_2 and B_3 of G satisfying conditions (1) and (2) of Theorem 1, in which B_1 and B_2 are simple, but B_3 is composed, since $B_1 < B_3$, $B_2 < B_3$ and $B_3 = B_1 + B_2$.

Theorem 2. *Let G be a directed graph. The following holds:*

(1) $\mathcal{G}(G)$ is a generator set for all the bubbles of G;
(2) $|\mathcal{G}(G)| \leq nm$.

Proof. (1) Recall that $\mathcal{S}(G)$ is the set of simple bubbles. By Theorem 1, $\mathcal{S}(G) \subseteq \mathcal{G}(G)$, and thus $\mathcal{G}(G)$ is a generator set for all the bubbles of G.
(2) Since every bubble b in $\mathcal{G}(G)$, with $\ell(b) = s, u_1, \ldots, t$ and $\mathcal{L}(b) = s, v_1, \ldots, v_r, t$, can be uniquely identified by its vertex s and its arc (v_r, t), then the number of bubbles in $\mathcal{G}(G)$ is upper-bounded by nm. ■

Remark 2. The upper bound given in Theorem 2 is asymptotically tight, as shown by the family of simple directed graphs on vertex set $V_n = \{1, 2, \ldots, n\}$ and all possible $n(n-1)$ arcs in their arc set $A_n = \{(u, v) : u \neq v, u, v \in V\}$.

Remark 3. Given a directed graph G, a naive algorithm to find $\mathcal{G}(G)$ would consist of the following steps. We start with $\mathcal{G}(G)$ as an empty set. We then find all-pairs shortest paths in G (since G is unweighted, this can be done through n BFSs). Finally, denoting, for each vertex $s \in V(G)$ and each arc $(v_r, t) \in A(G)$, by p_1 the shortest path from s to t in G and by p_2 the shortest path from s to v_r in G concatenated with the arc (v_r, t), we add the bubble $b = (p_1, p_2)$ to $\mathcal{G}(G)$ if p_1 and p_2 are internally vertex disjoint. Note that if $s = t$, then b corresponds to a degenerate bubble. A naive implementation of this algorithm takes $\mathcal{O}(n^2 m)$ time.

4 A polynomial-time algorithm for decomposing a bubble

The main result of this section is to provide a polynomial-time algorithm for decomposing any bubble of G into bubbles of $\mathcal{G}(G)$. To do so, we make use of a

tree-like decomposition. We need to take extra care in this decomposition since a naive approach could generate (several times) all the bubbles that are smaller than B, yielding an exponential number of steps.

Definition 4. *A bubble B is* short *if it satisfies condition (1) of Theorem 1, but not necessarily condition (2). Namely, let $\mathcal{L}(B) = s, v_1, \ldots, v_r, t$ be such that $\ell(B)$ is the shortest path from s to t in G but $s, v_1, \ldots, v_r$ is not necessarily the shortest path from s to v_r in G.*

We next introduce a measure for describing how "close" is a bubble to being short:

Definition 5. *Given an (s,t)-bubble B, let p^* be the shortest path from s to t. We say that B is k-*short*, for $k \geq 0$, if there is a leg $p \in \{\ell(B), \mathcal{L}(B)\}$ for which p^* and p share a prefix of exactly k arcs.*

Since in our case shortest paths are unique, only one leg of a bubble B can share a prefix with the shortest path p^*. Furthermore, any bubble B is k-short for some k, $0 \leq k \leq |\ell(B)|$. In particular, a bubble is short if and only if it is k-short for $k = |\ell(B)|$.

Definition 6. *Given a k-short bubble, we define the* short residual of B *as follows: $residual_s(B) = |B| - k$.*

Since $0 \leq k \leq |\ell(B)|$, and $|B| = |\ell(B)| + |\mathcal{L}(B)|$, we have that $|\mathcal{L}(B)| \leq residual_s(B) \leq |B|$.

We now present our polynomial time algorithm for decomposing a bubble of the graph G into bubbles of $\mathcal{G}(G)$. In the following, we assume that we have done a preprocessing step to compute all-pairs shortest paths in G in $\mathcal{O}(n(m+n))$ time through n BFSs.

Lemma 1. *Let B be an (s,t)-bubble that is not short. Then, B can be decomposed into two bubbles B_1 and B_2 ($B = B_1 + B_2$), such that: (a) B_1 is short, and (b) $residual_s(B_2) < residual_s(B)$. Moreover, B_1 and B_2 can be found in $\mathcal{O}(n)$ time.*

Proof. Let B be a k-short (s,t)-bubble, $0 \leq k < |\ell(B)|$. Let $p^* : s = x_0, x_1, \ldots, x_h = t$ be the shortest path from s to t in G. To prove (a), we follow a similar approach to Theorem 1. Since B is k-short, there is a leg $p \in \{\ell(B), \mathcal{L}(B)\}$ such that p^* and p share a prefix of exactly k arcs, $0 \leq k < h$. In other terms, leg p starts with arcs (x_0, x_1), ..., (x_{k-1}, x_k), the arc (x_k, x_{k+1}) is not in leg p, *i.e.*, x_k is the first vertex where the shortest path p^* departs from the leg p. Note that as a special case, $k = 0$ and $x_k = x_0 = s$. Let $l, k < l \leq h$, be the smallest index such that $x_l \in V(B)$. Namely, x_l is the first vertex after x_k where the shortest path p^* intersects again the bubble B. Such a vertex x_l must always exist, since $x_h = t \in V(B)$. Since $k < l$, we have that $|p^*_{k,l}| \geq 1$. We have two possible cases: either the vertices x_k and x_l are on the same leg of B (see Fig. 1(a)) or x_k and x_l are on different legs of B (see Fig. 1(b)). In either case, we can decompose

B as $B = B_1 + B_2$, as illustrated in Fig. 1. Note that in both cases, the bubble B_1 is short since one leg of B_1 is a subpath of the shortest path p^*, and hence a shortest path itself by subpath optimality.

Consider now B_2 in *Fig.* 1. To prove (b), we distinguish among the following three cases: (1) $x_k \neq s$ and vertices x_k and x_l are on the same leg of B; (2) $x_k \neq s$ and vertices x_k and x_l are on different legs of B; (3) $x_k = s$. First, consider case (1) (see Fig. 1(a)) and note that $residual_s(B) = |p_{k,l}|+|p_{l,h}|+|q_{0,h}|$ where q is the other leg of B different from p. Moreover, $residual_s(B_2) = |p_{l,h}| + |q_{0,h}|$. Hence, $residual_s(B) - residual_s(B_2) = |p_{k,l}| \geq |p^*_{k,l}| \geq 1$. Consider now case (2), (see Fig. 1(b)) and note that $residual_s(B) = |p_{0,l}|+|p_{l,h}|+|q_{k,h}|$ and $residual_s(B_2) = |p_{l,h}|+|q_{k,h}|$, and thus $residual_s(B) - residual_s(B_2) = |p_{0,l}| \geq |p^*_{0,k}|+|p^*_{k,l}| \geq 1$. The proof of case (3) is completely analogous to case (1), with $x_k = s$ and $p^*_{0,k} = \emptyset$, and again $residual_s(B) - residual_s(B_2) = |p_{k,l}| \geq |p^*_{k,l}| \geq 1$. In all cases, $residual_s(B) - residual_s(B_2) > 0$, and thus the claim follows. Finally, note that in order to compute B_1 and B_2 from B, it is sufficient to trace the shortest path p^*. Since all shortest paths are pre-computed in a preprocessing step, this can be done in $\mathcal{O}(n)$ time. ■

Lemma 2. *Any bubble B can be represented as a sum of $\mathcal{O}(n)$ (not necessarily distinct) short bubbles. This decomposition can be found in $\mathcal{O}(n^2)$ time in the worst case.*

Proof. Each time we apply Lemma 1 to a bubble B, we produce in $\mathcal{O}(n)$ time a short bubble B_1 and a bubble B_2 such that $residual_s(B_2) < residual_s(B)$. Since $residual_s(B) \leq |B| \leq n$, the lemma follows. ■

We next show how to further decompose short bubbles. Before doing that, we define the notion of *residual* for short bubbles, which measures how "close" is a short bubble to being a bubble of our generator set $\mathcal{G}(G)$.

Definition 7. *Let B be a short (s,t)-bubble, let $\ell(B) = p^*_1$ be the shortest path from s to t in G, let $\mathcal{L}(B) = s, v_1, \ldots, v_r, t$ be the other leg of B, let p^*_2 be the shortest path from s to v_r in G, and let p be the longest common prefix between $\mathcal{L}(B) - (v_r, t)$ and p^*_2. Then, the* residual *of B is defined as $residual(B) = |\mathcal{L}(B)| - 1 - |p|$.*

Since p is a prefix of $\mathcal{L}(B) - (v_r, t)$, we have that $0 \leq |p| \leq |\mathcal{L}(B)| - 1$. Thus, $0 \leq residual(B) \leq |\mathcal{L}(B)| - 1$.

Lemma 3. *Let B be a short (s,t)-bubble such that $residual(B) > 0$. B can be decomposed into two bubbles B_1 and B_2 ($B = B_1 + B_2$) such that B_1 and B_2 are short and $residual(B_1) + residual(B_2) < residual(B)$. Moreover, it is possible to find the bubbles B_1 and B_2 in $\mathcal{O}(n)$ time.*

Proof. Since B is a short (s,t)-bubble, it satisfies condition (1) of Theorem 1. Furthermore, as $residual(B) > 0$, it does not satisfy condition (2). Therefore, there exists two bubbles $B_1 < B$ and $B_2 < B$ such that $B = B_1 + B_2$ (from the proof of Theorem 1). Since $\ell(B)$ is the shortest path from s to t, using

arguments similar to the ones in Theorem 1, it can be shown that B can be decomposed into B_1 and B_2 and the only possible cases are the ones depicted in Fig. 2. Note that in all three cases of Fig. 2, each of the bubbles B_1 and B_2 has one leg that is a shortest path. Thus, in all three cases, B_1 and B_2 are short. Moreover, in Fig. 2(a), $residual(B_1) \leq |p_{l,h}| - 1$ and $residual(B_2) \leq |p_{k,l}| - 1$. Therefore, $residual(B_1) + residual(B_2) \leq |p_{l,h}| - 1 + |p_{k,l}| - 1 = residual(B) - 1 < residual(B)$. Similarly, in Fig. 2(b) and (b_1), $residual(B_1) \leq |p_{0,l}| - 1$, $residual(B_2) \leq |p_{l,h}| - 1$, and thus, $residual(B_1) + residual(B_2) \leq |p_{0,l}| - 1 + |p_{l,h}| - 1 = residual(B) - 1 < residual(B)$. In all three cases, B_1 and B_2 are short and $residual(B_1) + residual(B_2) < residual(B)$. The claim thus follows.

Once again, observe that in order to compute B_1 and B_2 from B, it is sufficient to trace the shortest path from s to t. Since all shortest paths are precomputed in a preprocessing step, this can be done in $\mathcal{O}(n)$ time. ■

Lemma 4. *Any short bubble B has a tree-like decomposition into $\mathcal{O}(n)$ (not necessarily distinct) bubbles from the generator $\mathcal{G}(G)$. This decomposition can be found in $\mathcal{O}(n^2)$ time in the worst case.*

Proof. Each time we apply Lemma 3 to a short bubble B, we produce in $\mathcal{O}(n)$ time two short bubbles B_1 and B_2 such that $residual(B_1) + residual(B_2) < residual(B)$. Since $|\ell(B)| + residual(B) \leq n$, this implies that a short bubble can be decomposed in $\mathcal{O}(n)$ bubbles from the generator set $\mathcal{G}(G)$ in $\mathcal{O}(n^2)$ time. ■

Theorem 3. *Given a graph G, any bubble B in G can be represented as a sum of $\mathcal{O}(n^2)$ bubbles that belong to $\mathcal{G}(G)$. This decomposition can be found in a total of $\mathcal{O}(n^3)$ time.*

Proof. The theorem follows by Lemma 2 and Lemma 4. ■

5 Conclusions and open problems

Bubbles in de Bruijn graphs represent interesting biological events, like alternative splicing and allelic differences (SNPs and indels). However, the set of all bubbles in a de Bruijn graph built from real data is usually too large to be efficiently enumerated and analysed. Therefore, in this paper we proposed a bubble generator, which is a polynomial-sized subset of the bubble space that can be used to generate all and only the bubbles in a directed graph. The concept of bubble generators is similar to cycle bases, but the algorithms for the latter cannot be applied as black boxes to find the former because the bubble space does not form a vector space. As such, this work describes efficient algorithms to identify, for any given directed graph G, a generator set of bubbles $\mathcal{G}(G)$, and to decompose a given bubble B into bubbles from $\mathcal{G}(G)$.

There remain several theoretical open questions. First, our generator $\mathcal{G}(G)$ is not necessarily minimal, *i.e.* it might happen that there exists three bubbles $B_1, B_2, B_3 \in \mathcal{G}(G)$ such that $B_1 < B_3$, $B_2 < B_3$, and $B_3 = B_1 + B_2$. Is it possible to find in polynomial time a generator $\mathcal{G}'(G)$ that is minimal or even better,

to find $\mathcal{S}(G)$? Second, it would be interesting to know if there are polynomial-time algorithms to decompose any bubble of a graph G into bubbles of such generators. Third, it would be interesting to find a generator $\mathcal{G}(G)$ with some additional biologically motivated constraints, such as for example on the maximum length of the legs of a bubble [15]. Given an integer k and a graph G, is it possible to find a generator $\mathcal{G}(G)$ that generates all and only the bubbles of G which have both legs of length at most k? Fourth, are there faster algorithms to find a bubble generator? Fifth, this work is related to the research done in the direction of cycle bases. However, as we already mentioned, our problem displays characteristics that make it very different from the ones related to cycle bases. Thus, it may be of independent interest to further investigate the connections between these problems.

Finally, application of the bubble generator to genomic and transcriptomic graphs must be explored since it is one of the main motivations for this theoretical study. Similarly to the case of cycle bases, the simplest application of the bubble generators is to use it as a preprocessing step in several algorithms to reduce the amount of work to be done. For example, it can remove from the graph all unnecessary arcs (*i.e.* arcs that do not belong to any bubble) in order to lower the running time of an algorithm that is only interested in bubbles. As another example, the polynomial-time decomposition algorithm can be useful in the case where we want to identify and decompose complex alternative splicing events [17] into their elementary parts. However, exploring possible applications of the bubble generator is out of the scope of this paper.

Acknowledgments

V. Acuña was supported by Fondecyt 1140631, CIRIC-INRIA Chile and Basal Project PBF 03. R. Grossi and G. F. Italiano were partially supported by MIUR, the Italian Ministry of Education, University and Research, under the Project AMANDA (Algorithmics for MAssive and Networked DAta). Part of this work was done while G. F. Italiano was visiting Université de Lyon. L. Lima is supported by the Brazilian Ministry of Science, Technology and Innovation (in portuguese, Ministério da Ciência, Tecnologia e Inovação - MCTI) through the National Counsel of Technological and Scientific Development (in portuguese, Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq), under the Science Without Borders (in portuguese, Ciências Sem Fronteiras) scholarship grant process number 203362/2014-4. B. Sinaimeri, L. Lima and M.-F. Sagot are partially funded by the French ANR project Aster (2016-2020), and together with V. Acuña, also by the Stic AmSud project MAIA (2016-2017). This work was performed using the computing facilities of the CC LBBE/PRABI.

References

1. Birmelé, E., Crescenzi, P., Ferreira, R., Grossi, R., Lacroix, V., Marino, A., Pisanti, N., Sacomoto, G., Sagot, M.F.: Efficient Bubble Enumeration in Directed Graphs. In: SPIRE. pp. 118–129 (2012)

2. Bollobás, B.: Modern graph theory, Graduate Texts in Mathematics, vol. 184. Springer-Verlag, Berlin (1998)
3. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications. Elsevier, New York (1976)
4. Brankovic, L., Iliopoulos, C.S., Kundu, R., Mohamed, M., Pissis, S.P., Vayani, F.: Linear-time superbubble identification algorithm for genome assembly. Theoretical Computer Science 609, 374–383 (2016)
5. Deo, N.: Graph theory with applications to engineering and computer science. Prentice-Hall series in automatic computation, Englewood Cliffs, N.J. Prentice-Hall (1974)
6. Gleiss, P.M., Leydold, J., Stadler, P.F.: Circuit bases of strongly connected digraphs. Discussiones Mathematicae Graph Theory 23(2), 241–260 (2003)
7. Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., McVean, G.: De novo assembly and genotyping of variants using colored de bruijn graphs. Nat Genet 44(2), 226–232 (2012)
8. Kavitha, T., Liebchen, C., Mehlhorn, K., Michail, D., Rizzi, R., Ueckerdt, T., Zweig, K.A.: Cycle bases in graphs characterization, algorithms, complexity, and applications. Computer Science Review 3(4), 199 – 243 (2009)
9. Kavitha, T., Mehlhorn, K.: Algorithms to compute minimum cycle bases in directed graphs. Theory of Computing Systems 40(4), 485 – 505 (2007)
10. Li, H.: Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. Bioinformatics 28(14), 1838–1844 (2012)
11. Lima, L., Sinaimeri, B., Sacomoto, G., Lopez-Maestre, H., Marchet, C., Miele, V., Sagot, M.F., Lacroix, V.: Playing hide and seek with repeats in local and global de novo transcriptome assembly of short RNA-seq reads. Algorithms for molecular biology 12 (2017)
12. MacLane, S.: A combinatorial condition for planar graphs. Fundamenta Mathematicae 28, 22–32 (1937)
13. Onodera, T., Sadakane, K., Shibuya, T.: Detecting Superbubbles in Assembly Graphs. In: Algorithms in Bioinformatics, Lecture Notes in Computer Science, vol. 8126, pp. 338–348. Springer Berlin Heidelberg (2013)
14. Pevzner, P.A., Tang, H., Tesler, G.: De Novo Repeat Classification and Fragment Assembly. Genome Research 14(9), 1786–1796 (2004)
15. Sacomoto, G., Lacroix, V., Sagot, M.F.: A polynomial delay algorithm for the enumeration of bubbles with length constraints in directed graphs and its application to the detection of alternative splicing in RNA-seq data. In: WABI. pp. 99–111 (2013)
16. Sacomoto, G., Kielbassa, J., Chikhi, R., Uricaru, R., Antoniou, P., Sagot, M.F., Peterlongo, P., Lacroix, V.: Kissplice: de-novo calling alternative splicing events from rna-seq data. BMC Bioinformatics 13(S-6), S5 (2012)
17. Sammeth, M.: Complete alternative splicing events are bubbles in splicing graphs. Journal of Computational Biology 16(8), 1117–1140 (2009)
18. Simpson, J.T., Wong, K., Jackman, S.D., Schein, J.E., Jones, S.J.M., Birol, I.: ABySS: A parallel assembler for short read sequence data. Genome Research 19(6), 1117–1123 (2009)
19. Sung, W.K., Sadakane, K., Shibuya, T., Belorkar, A., Pyrogova, I.: An o(m log m)-time algorithm for detecting superbubbles. IEEE/ACM Trans. Comput. Biol. Bioinformatics 12(4), 770–777 (2015)
20. Uricaru, R., Rizk, G., Lacroix, V., Quillery, E., Plantard, O., Chikhi, R., Lemaitre, C., Peterlongo, P.: Reference-free detection of isolated SNPs. Nucleic Acids Research 43(2), e11 (2015)

21. Younsi, R., MacLean, D.: Using 2k+2 bubble searches to find single nucleotide polymorphisms in k-mer graphs. Bioinformatics 31(5), 642–646 (2015)
22. Zerbino, D., Birney, E.: Velvet: Algorithms for De Novo Short Read Assembly Using De Bruijn Graphs. Genome Res. (2008)