# Telling stories: Enumerating maximal directed acyclic graphs with a constrained set of sources and targets

Vicente Acuña, Etienne Birmelé, Ludovic Cottret, Pierluigi Crescenzi, Fabien Jourdan, Vincent Lacroix, Alberto Marchetti-Spaccamela, Andrea Marino, Paulo Vieira Milreu, Marie-France Sagot, et al.

# Telling Stories

Vicente Acuña[1,2], Etienne Birmelé[3], Ludovic Cottret[4], Pierluigi Crescenzi[4], Fabien Jourdan[5], Vincent Lacroix[1,2], Alberto Marchetti-Spaccamela[6], Andrea Marino[4], Paulo Vieira Milreu[1], Marie-France Sagot[1,2], and Leen Stougie[7]

[1] Université de Lyon, F-69000 Lyon; Université Lyon 1; CNRS, UMR5558, Laboratoire de Biométrie et Biologie Evolutive, F-69622 Villeurbanne, France
[2] INRIA Rhône-Alpes, 38330 Montbonnot Saint-Martin, France
[3] Lab. Statistique et Génome, CNRS UMR8071 INRA1152, Université d'Évry, France
[4] Laboratoire d'Ingénierie des Systèmes Biologiques et des Procédés (LISBP), UMR CNRS 5504 - INRA 792, Toulouse, France
[5] INRA, UMR1089 Xénobiotiques, Toulouse, France
[6] Università di Firenze, Dipartimento di Sistemi e Informatica, I-50134 Firenze, Italy
[7] Sapienza University of Rome, Italy,
[8] VU University and CWI, Amsterdam, The Netherlands

**Abstract.** We present in this paper a constrained version of the problem of enumerating all maximal directed acyclic subgraphs (DAG) of a graph $G$. In this version, we enumerate stories, which are maximal DAGs whose sources and targets belong to a predefined subset of the nodes. First we show how to compute one story in polynomial-time, and we then describe two different algorithms to "tell" all possible stories.

## 1 Introduction

We present in this paper a constrained version of the problem of enumerating all maximal directed acyclic subgraphs (DAG) of a graph $G$ [5]. In this version, only a given subset $\mathbb{B}$ of the nodes are allowed to be sources or sinks of the DAGs that have to be enumerated. This problem was motivated initially by a biological question [4] related to metabolic networks: in these networks, nodes represent chemical compounds and an arc between two nodes $u$ and $v$ indicates that $v$ can be obtained by a chemical transformation of $u$ (plus possibly of some other compound(s)) via a given metabolic reaction [3]. The subset $\mathbb{B}$ corresponds to compounds that have been experimentally identified as having a significantly higher or lower production in a given condition (for instance when an organism is exposed to some stress). The aim is then to extract all the interaction dependencies among the compounds in $\mathbb{B}$ which do not create cycles but at the same time involve as many compounds as possible. These may require intermediate steps that concern compounds not in $\mathbb{B}$ but the initial and final steps must involve only compounds in $\mathbb{B}$. A solution, that is a possible scenario of metabolic dependencies, is called a *(metabolic) story*. The problem is then to "tell" all possible stories given as input a graph $G$ and a subset $\mathbb{B}$ of the nodes of $G$.

Although the problem was originally motivated by biology and seems to be related to a Steiner problem, it is surprising that, as far as we know, such constraint on sources and sinks was never considered before. In this paper, we show that this constraint is enough to change the nature of the enumeration problem. Indeed, this can now be seen as a generalisation of the feedback arc set problem (FAS) since the complement of a story is a minimal set of arcs that breaks all the cycles and also avoids sources or sinks that are not in $\mathbb{B}$. We call such minimal sets of arcs *story arc sets (SAS)*. We show that the two notions, FAS and SAS, are however different, and that telling stories is possibly harder than enumerating feedback arc sets.

This paper is organised in the following way. The next section presents the main definitions and notations. Section 3 presents some operations to simplify the graph without losing solutions. Sections 4 and 5 propose two different approaches to enumerate stories: the first one makes use of a minimal feedback-arc-set enumerator but it can only be applied to a specific class of graphs while the second one is an extension of our algorithm to enumerate one story based on an initial permutation of the nodes and can be used for any graph. Finally, Section 6 shows that the problem of finding stories with a specific set of sources and sinks is NP-complete.

## 2 Preliminaries

Let $G = (\mathbb{B} \cup \mathbb{W}, E)$ be a directed graph such that $\mathbb{B} \cap \mathbb{W} = \emptyset$. Nodes in $\mathbb{B}$ are said to be *black* while those in $\mathbb{W}$ are said to be *white*. Given a node $u$, the in-degree of $u$ corresponds to the number of arcs incoming to $u$ while the out-degree of $u$ corresponds to the number of arcs outgoing from $u$. A node of a directed graph is said to be a source (respectively, a sink) if its in-degree is 0 and its out-degree is positive (respectively, its in-degree is positive and its out-degree is 0).

A **pitch** of $G$ is an acyclic subgraph $G' = (\mathbb{B} \cup \mathbb{W}', E')$ of $G$ where $\mathbb{W}' \subseteq \mathbb{W}$ and $E' \subseteq E$ and, for each node $w \in \mathbb{W}'$, the in-degree and the out-degree of $w$ are both greater or equal to 1. A trivial pitch can be obtained with $\mathbb{W}' = \emptyset$ and $E' = \emptyset$, i.e, the subgraph containing all the black nodes and no arc. We define a **story** as a maximal pitch: the story problem consists in enumerating all possible stories. In the following, we will denote by $\Sigma(G)$ the set of stories of $G$.

> **Problem** ENUM-STORIES($G$): given a directed graph $G = (\mathbb{B} \cup \mathbb{W}, E)$ such that $\mathbb{B} \cap \mathbb{W} = \emptyset$, enumerate all maximal directed acyclic subgraphs of $G$ containing as sources and targets only nodes of $\mathbb{B}$.

A **feedback arc set** (FAS) of a directed graph $G = (V, E)$ is a subset $F$ of $E$ such that $G_{F^-} = (V, E \setminus F)$ is acyclic. A FAS is said to be *minimal* if there exists no $f \in F$ such that $F \setminus \{f\}$ is a FAS. One could think of the complement of a FAS as a story but this is not the case since $G_{F^-}$ can contain white sources or sinks. Indeed, the FAS problem is a particular instance of our problem in which every node is black, i.e, $\mathbb{W} = \emptyset$. Analogously, we can define the **story**

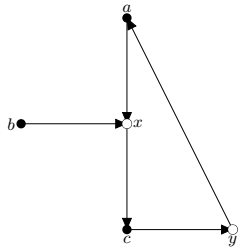


Fig. 1: In this case, $\mathbb{B} = \{a, b, c\}$ and $\mathbb{W} = \{x, y\}$. There are 4 possible minimal FASs, that is, $\{(a, x)\}$, $\{(x, c)\}$, $\{(c, y)\}$, and $\{(y, a)\}$. Only one of these minimal FASs (that is, the first one) is also a minimal SAS. For example, the second one is not a SAS since $G_{\{(x,c)\}^-}$ contains a white sink (that is, node $x$). On the other hand, another minimal SAS is $\{(c, y), (y, a)\}$, which is not a minimal FAS (even though it is a FAS).

**arc set** (SAS), which is a FAS $S$ such that no white node in $G_{S^-}$ is a source or a sink. A SAS is said to be *minimal* if there exists no subset $S'$ of $S$ such that $S \setminus S'$ is a SAS. This implies that if $S$ is minimal, then for every $s \in S$, the graph $G_{S^-,s^+} = (\mathbb{B} \cup \mathbb{W}, E \setminus S \cup \{s\})$ either contains a cycle or contains a white source or sink. If $S$ is a minimal SAS, then $G_{S^-}$ is a story. A SAS is also a FAS. However, the example presented in Figure 1 shows that, in general, a minimal FAS is not a minimal SAS and that a minimal SAS is not a minimal FAS. For this reason the use of a polynomial-time delay enumeration algorithm for minimal FAS as the one proposed in [5] to enumerate stories is limited, since some minimal SAS may not be detected. We shall see in a later section that this is not the case when we restrict ourselves to some particular class of graphs.

## 3 A Simple Preprocessing of the Graph

In this section, we show how the graph may be simplified without changing the set of its stories: this simplification will allow us to make the proofs of our results shorter. Moreover, the simplified graphs turn out to be interesting even from a biological point of view, since they are a more compact representation of graphs equivalent in terms of story sets.

In particular, we will use the following four simplification operations.

- A **dead-end removal** consists in removing (1) white nodes that cannot be reached by any of the black nodes and (2) white nodes that cannot reach any of the black nodes. Let $\mathtt{de}(G)$ be the resulting graph. Notice that such white nodes cannot be part of any story, since they belong to no path between two black nodes. Hence, $\Sigma(G) = \Sigma(\mathtt{de}(G))$. Observe also that, in the graph obtained after the application of this operation, all white nodes are intermediate, i.e, there is no white source or sink.

– **Self-loop removal** consists in removing all arcs of the form $(u, u)$. Let $\mathtt{sl}(G)$ be the resulting graph. Self-loops are cycles of only one arc and, since stories are acyclic, this arc cannot be part of any story. Thus, $\Sigma(G) = \Sigma(\mathtt{sl}(G))$.
– A **forward bottleneck removal** consists in removing a white node $v$ whose out-degree is equal to 1, and in directly connecting any predecessor of $v$ to the unique successor of $v$ (without creating multi-arcs). Let $\mathtt{fb}(G, v)$ be the resulting graph.
– A **backward bottleneck removal** consists in removing a white node whose in-degree is equal to 1, and in directly connecting the unique predecessor of $v$ to the successors of $v$ (without creating multi-arcs). Let $\mathtt{bb}(G, v)$ be the resulting graph.

Observe that, when applying the last three operations, the number of arcs of the resulting graphs is always smaller than the number of arcs of the original graph.

**Lemma 1.** *Let $v$, $p$, and $s$ be three nodes such that $(p, v), (v, s), (p, s) \in E$. Then, for any story $S$, $(p, v), (v, s) \in S$ if and only if $(p, s) \in S$.*

*Proof.* Assume, on the contrary, that $(p, v), (v, s) \in S$ but $(p, s) \notin S$. This contradicts the maximality of $S$. Indeed, if we add $(p, s)$ to $S$, then we cannot create a new cycle since the two arcs $(p, v)$ and $(v, s)$ belong to the same cycle $C$ of $G$ if and only if the arc $(p, s)$ belongs to the cycle $C'$ of $G$ which contains all the arcs of $C$ but $(p, v)$ and $(v, s)$. Moreover, adding $(p, s)$ to $S$ cannot create a white source (that is, $p$ if $p$ is white) since in this case $p$ would be already a source in $S$, and it cannot create a white target (that is, $s$ if $s$ is white) since in this case $s$ would be already a target in $S$. Similarly, we can prove that if $(p, s) \in S$ then $(p, v), (v, s) \in S$. □

Given three nodes $v$, $p$, and $s$ such that $(p, v), (v, s) \in E$ and $(p, s) \notin E$, let $\mathtt{ab}(G, v, p, s)$ denote the graph obtained by adding to $G$ the arc $(p, s)$.

**Lemma 2.** *Let $v$ be a forward bottleneck and let $p$ and $s$ be two nodes such that $(p, v), (v, s) \in E$ and $(p, s) \notin E$. Then $\Sigma(G) = \Sigma(\mathtt{ab}(G, v, p, s))$.*

*Proof.* For any story $S$ of $G$, we define $f(S) = S \cup \{(p, s)\}$ if $(p, v) \in S$ (and, hence, $(v, s) \in S$), otherwise $f(S) = S$. Let us first prove that $S' = f(S)$ is a story of $G' = \mathtt{ab}(G, v, p, s)$. Since the arc $(p, s)$ belongs to a cycle $C'$ of $G'$ if and only if the two arcs $(p, v)$ and $(v, s)$ belong to the cycle $C$ of $G$ which contains all the arcs of $C'$ but $(p, s)$, we have that $S$ is acyclic if and only if $S'$ is acyclic. Moreover, $S'$ is maximal. Indeed, if $(p, s) \in S'$, then any other set of arcs could not be added to $S'$ since otherwise it could be added to $S$. Otherwise, if $(p, s)$ could be added to $S'$, then, from Lemma 1 also $(p, v)$ and $(v, s)$ could be added to $S'$ and, hence, these two arcs could be added to $S$. Let us now prove that, if $S_1$ and $S_2$ are two stories such that $S_1 \neq S_2$, then $f(S_1) \neq f(S_2)$. If $(p, v) \notin S_1 \cup S_2$, then $f(S_1) = S_1 \neq S_2 = f(S_2)$. Otherwise, if $(p, v) \in S_1 \cap S_2$, then $f(S_1) = S_1 \cup \{(p, s)\} \neq S_2 \cup \{(p, s)\} = f(S_2)$. Finally, if $(p, v) \in S_1 - S_2$ (the

other case can be dealt with similarly), then $(p, s) \in f(S_1)$ while $(p, s) \notin f(S_2)$ and, hence, $f(S_1) \neq f(S_2)$. It then remains to show that, for any story $S'$ of $G'$, there exists a story $S$ of $G$ such that $f(S) = S'$. Let us define $S = S' - \{(p, s)\}$: clearly, $S$ is acyclic (since it is a subgraph of an acyclic graph). If $(p, s) \notin S'$, then $S = S'$ is a story in $G$ (since the only difference between $G$ and $G'$ is the arc $(p, s)$). Otherwise, from Lemma 1 it follows that $(p, v), (v, s) \in S'$ and, hence, $(p, v), (v, s) \in S$: the maximality of $S$ then follows from the maximality of $S'$, since any set of arcs that could be added to $S$ could also be added to $S'$. □

According to the previous lemma, we can now assume that, for any forward bottleneck $v$ whose unique successor is $s$ and for any predecessor $p$ of $v$, the graph contains the arc $(p, s)$. Given three nodes $v$, $p$, and $s$ such that $(p, v), (v, s), (p, s) \in E$, let $\mathtt{dp}(G, v, p, s)$ denote the graph obtained by removing from $G$ the two arcs $(p, v)$ and $(v, s)$.

**Lemma 3.** *Let $v$ be a forward bottleneck and let $p$ and $s$ be two nodes such that $(p, v), (v, s), (p, s) \in E$. Then $\Sigma(G) = \Sigma(\mathtt{dp}(G, v, p, s))$.*

*Proof.* For any story $S$ of $G$, we define $f(S) = S - \{(p, v), (v, s)\}$. Let us first prove that $S' = f(S)$ is a story of $G' = \mathtt{dp}(G, v, p, s)$. Clearly, $S'$ is acyclic (since it is a subgraph of an acyclic graph). Moreover, from Lemma 1 it follows that if $(p, v), (v, s) \in S$, then $(p, s) \in S$ and, hence, $(p, s) \in S'$: the maximality of $S'$ then follows from the maximality of $S$, since any set of arcs that could be added to $S'$ could also be added to $S$. Let us now prove that, if $S_1$ and $S_2$ are two stories such that $S_1 \neq S_2$, then $f(S_1) \neq f(S_2)$. If $(p, s) \notin S_1 \cup S_2$, then $(p, v), (v, s) \notin S_1 \cup S_2$ and $f(S_1) = S_1 \neq S_2 = f(S_2)$. Otherwise, if $(p, s) \in S_1 \cap S_2$, then $(p, v), (v, s) \in S_1 \cap S_2$ and $f(S_1) = S_1 - \{(p, v), (v, s)\} \neq S_2 - \{(p, v), (v, s)\} = f(S_2)$. Finally, if $(p, s) \in S_1 - S_2$ (the other case can be dealt with similarly), then $(p, s) \in f(S_1)$ while $(p, s) \notin f(S_2)$ and, hence, $f(S_1) \neq f(S_2)$. It then remains to show that, for any story $S'$ of $G'$, there exists a story $S$ of $G$ such that $f(S) = S'$. Let us define $S = S' \cup \{(p, v), (v, s)\}$ if $(p, s) \in S'$, otherwise $S = S'$. Similarly to what we have done at the beginning of the proof of the previous lemma, we can prove that $S$ is a story. □

From the above two lemmas, the next result immediately follows.

**Theorem 4.** *For any forward bottleneck $v$, $\Sigma(G) = \Sigma(\mathtt{fb}(G, v))$.*

Analogously we can prove the following result which concerns the removal of backward bottlenecks.

**Theorem 5.** *For any backward bottleneck $v$, $\Sigma(G) = \Sigma(\mathtt{bb}(G, v))$.*

For any graph $G$, let $\mathtt{fb}(G)$ (respectively $\mathtt{bb}(G)$) denote the graph obtained by applying as many times as possible the forward (respectively backward) bottleneck removal operation. Notice that, even if $G$ does not contain self-loops, it might happen that $\mathtt{fb}(G)$ (respectively $\mathtt{bb}(G)$) contains self-loops created by one bottleneck removal. Our simplification procedure can now be described as follows.

1. Let $G_0 = \texttt{sl}(\texttt{de}(G))$ and let $i = 0$.
2. Let $G_{i+1} = \texttt{sl}(\texttt{bb}(\texttt{sl}(\texttt{fb}(G_i))))$.
3. If $G_{i+1} = G_i$ then return $G_i$, otherwise let $i = i + 1$ and go to Step 2.

As a consequence of the previous results, we have that if $H$ is the graph returned by this procedure, then $\Sigma(G) = \Sigma(H)$. In the following, we can now assume that any white node has both in-degree and out-degree greater than 1. Notice that this avoids graphs like the one shown in Figure 1. Indeed, in this case, the two arcs $(c, y)$ and $(y, a)$ would disappear and the arc $(c, a)$ would be inserted. Observe also that this simplification procedure does not guarantee that a minimal FAS enumerator would produce all possible minimal SAS as we shall see in the next section.

## 4 Enumerating stories by enumerating FASs

We already noticed that there exists graphs for which the set $\mathcal{S}(G)$ of minimal SASs and the set $\mathcal{F}(G)$ of minimal FASs are not comparable in terms of the inclusion relation. In this section, we show that, for some particular cases, $\mathcal{S}(G)$ is contained in $\mathcal{F}(G)$. To this aim, let us introduce the following definition: a white node $v$ is **bad** if, for any predecessor $p$ of $v$ and for any successor $s$ of $v$, there exists a cycle containing the two arcs $(p, v)$ and $(v, s)$ (see Figure 2).

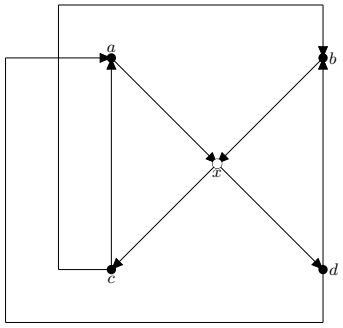


Fig. 2: Example of a bad node. The minimal SAS $\{(a, x), (b, x), (x, c), (x, d)\}$ is not a minimal FAS.

**Proposition 6.** *Let $v$ be a white node of $G$ which is not bad. Then $v$ belongs to any story.*

*Proof.* Consider a pitch $P$ not containing $v$. As $v$ is not bad, so there exist a predecessor $p$ of $v$ and a successor $s$ of $v$ such that there exist no cycle containing the two arcs $(p, v)$ and $(v, s)$. According to the results of the previous section, we can assume that there exists a path $p_k, p_{k-1}, \ldots, p_1 = p$ with $k \geq 2$ such that

$p_k$ is black and $p_i$ is white, for any $i$ with $1 \leq i < k$. Let $j$ be the minimum $i$ with $1 \leq i < k$ such that $p_i$ is in $P$: if no such $j$ exists, then we define $j = k$. A path $s = s_1, \ldots, s_{l-1}, s_l$ ending in a black node can be found in a similar way, and let $s_{j'}$ be the first node on that path belonging to $P$ or $s_{j'} = s_l$ if no such node exists.

Then $P' = P \cup \{(p_j, p_{j-1}), \ldots, (p, v), (v, s), \ldots, (s_{j'-1}, s_{j'})\}$ has no white source nor sink as $p_j$ and $s'_j$ are not white sources or sinks in $P$. Moreover, $P'$ is acyclic as $P$ is acyclic and any cycle containing the additional path would contradict the fact that $v$ is not a bad node. Thus any pitch not containing $v$ is not maximal, that is is not a story.

**Corollary 7.** *If $G$ does not include any bad node, then any minimal* SAS *is a minimal* FAS*.*

*Proof.* By absurdum, assume that $S$ is a minimal SAS which is not a minimal FAS. Then, there exists an arc $e = (u, v) \in S$ such that $S \cup \{e\}$ is a FAS but not a SAS, that is, in $G_{S^-, e^+}$ either $u$ is a white source or $v$ is a white sink. We can restrict ourselves to consider the latter case, since the former one can be dealt with in a similar way. Since in $G_{S^-, e^+}$ $v$ is a white sink, we have that all arcs incident to $v$ are in $S$. In other words, $G_S$ does not contain $v$, which contradicts Proposition 6.

The previous proposition and its corollary states that, in a graph with no bad nodes, each story corresponds to a minimal FAS. This suggests that we could enumerate all stories by enumerating all the minimal FASs and by checking for each of them whether the resulting graph is a story (which can be done by checking that no white node is source or sink). Unfortunately, there are graphs with no bad nodes in which the number of minimal FASs is exponentially larger than the number of minimal SASs. An example is given in Figure 3.

Fig. 3: Graph with no bad node and in which the number of minimal FASs is $2^n$ and the number of minimal SASs is 2.

## 5 Enumerating stories by enumerating permutations

In the previous section, we suggested a method for enumerating all stories in the case of graphs with no bad nodes. Unfortunately, many graphs arising from the biological application described in the introduction contain a huge number

of bad nodes. We thus need a method for enumerating stories which is able to deal with these cases.

Let us first consider the case of finding a first story. We now show that this can be done in polynomial time. Our algorithm basically starts with a pitch and grows it to a story by adding paths between black nodes while avoiding cycles. To this purpose, we can start with a trivial pitch such as the subgraph containing all the black nodes and no arcs.

---

Algorithm COMPLETE_PITCH$(G, P)$
**Require:** a graph $G = (\mathbb{B} \cup \mathbb{W}, E))$ with $\mathbb{B} \cap \mathbb{W} = \emptyset$ and an initial pitch $P$;
**Ensure:** A story completing $P$

```
i ← 1
π ← any topological order of P
while i ≤ |V(P)|) do
    u ← i-th element according to π with u ∈ V(P)
    Apply BFS(u, G \ E(P)) until reach a node v ∈ V(P)
    if π(u) < π(v) ∨ (u and v are uncomparable) then
        include the path u ⇝ v in P and update π
        i ← 1
    else if no such node v exists then
        i ← i + 1
return P
```

---

**Theorem 8.** *A story can be computed in polynomial time.*

*Proof.* Algorithm COMPLETE_PITCH computes a story by completing a starting pitch $P$. First of all, the algorithm computes a topological order $\pi$ of the nodes consistent with the pitch: let $u$ be the first node in this order. Successively, a path $u \rightsquigarrow v$ with $u, v \in V(P)$ and all arcs in $u \rightsquigarrow v$ not in $E(P)$ is found by applying a modified breadth-first-search starting with $u$ and traversing only arcs not in $P$, until a node $v$ in $V(P)$ is hit such that $\pi(u) < \pi(v)$ or $u$ and $v$ are incomparable. In this case, the path $u \rightsquigarrow v$ is added to $P$ and the topological order is updated. This path is guaranteed to create no cycle since there was no path $v \rightsquigarrow u$ in $P$ due to the fact that $\pi(u) < \pi(v)$. Moreover, since $P$ contained no white source nor sink before the addition of the path, then it does not contain them after such an addition because $u$ and $v$, which are the only candidates to become source or sink, were already present in $P$. Thus the addition of $u \rightsquigarrow v$ to $P$ creates a new pitch. This procedure is repeated until no new path starting from $u$ can be found. At this point, we continue with the next node in the updated order $\pi$ with the modification that, whenever a new path is found, the entire procedure is started again from the minimum node according to the order. Since at each updating of the topological order, we add at least one arc, the algorithm terminates in polynomial time. □

We now shift our attention to the problem of enumerating all stories. To this aim, we define two simple operations, CLEAN and CONSISTENT_ARCS as follows. For any graph $G(\mathbb{B} \cup \mathbb{W}, E)$ and for any total order $\pi$ of the nodes:

$G'(\mathbb{B} \cup \mathbb{W}, E') \equiv$ CONSISTENT_ARCS$(G, \pi)$: for each arc $(u, v) \in E$, $(u, v) \in E'$ if $\pi(u) < \pi(v)$;
$G'(\mathbb{B} \cup \mathbb{W}', E') \equiv$ CLEAN$(G)$: recursively remove white nodes that are sources, sinks or isolated in $G$.

We can thus define PITCH$(G, \pi)$ as CLEAN(CONSISTENT_ARCS$(G, \pi)$). PITCH produces a pitch since the resulting graph $G'$ contains only arcs that respect the order $\pi$ and therefore is acyclic. Moreover, due to the cleaning step, $G'$ is guaranteed to have neither white sources nor sinks.

**Theorem 9.** *For any story $S$, there exists a permutation $\pi$ such that* PITCH$(G, \pi) = S$.

*Proof.* In order to prove the theorem, it is enough to show that, for any story $S$ of $G = (\mathbb{B} \cup \mathbb{W}, E)$ and for any topological order $\pi$ of $V(S)$, PITCH$(G, \pi) = S$. To this aim, because of the maximality of a story, it suffices to show that $S \subseteq$ PITCH$(G, \pi)$). Given an arc $(u, v)$ of $S$, we have $\pi(u) < \pi(v)$. Therefore $(u, v)$ is in CONSISTENT_ARCS$(G, \pi)$. Since $(u, v)$ is an arc of $S$, there exists a path $p$ in $S$ between two black nodes containing $u$ and $v$. Then $p$ is also in CONSISTENT_ARCS$(G, \pi)$ and thus $u$ and $v$ are both black or, if one or both of them is white, then they are neither source nor sink in CONSISTENT_ARCS$(G, \pi)$. Since CLEAN(CONSISTENT_ARCS$(G, \pi)$) does not remove black nodes nor white nodes that are neither source nor sink, we can conclude then that $(u, v)$ is also in CLEAN(CONSISTENT_ARCS$(G, \pi)$) = PITCH$(G, \pi)$. □

The previous two theorems suggest an approach to enumerate stories which simply consists in generating all permutations $\pi$ of the nodes of $G$ and computing $P =$ PITCH$(G, \pi)$: if $P$ is not a story then we use COMPLETE_PITCH to make it a story.

## 6 Finding specific stories

As stated in the previous section, finding a story is polynomial. However the problem becomes NP-complete if we have to identify specific stories, i.e., ones having a particular set of sources and/or sinks.

**Theorem 10.** *Deciding whether there exists a story with a given set of sources and sinks is NP-complete.*

*Proof.* In order to prove this theorem, we show how the `3-SAT` problem[1] is reducible to the problem of deciding whether, given a directed graph $G = (V, E)$ and two subsets $S$ and $T$ of $V$, $G$ includes a maximal DAG whose set of sources (respectively, sinks) is equal to $S$ (respectively, $T$). To this aim, let us consider a 3-CNF Boolean formula $\varphi$ consisting of $m$ clauses $c_i$ over a set of $n$ Boolean variables $x_j$: we then define a directed graph $G$ as follows (see also Figure 4).

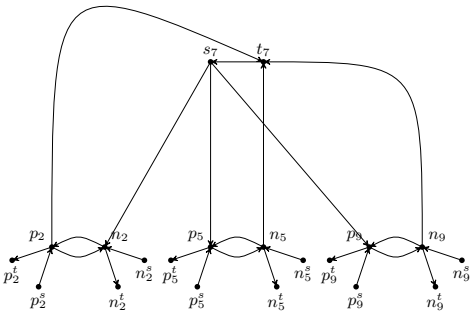


Fig. 4: The subgraph corresponding to the clause $c_7 = \neg x_2 \vee x_5 \vee x_9$

- The set of nodes of $G$ includes six nodes $p_j, p_j^s, p_j^t, n_j, n_j^s, n_j^t$ for each variable $x_j$, and two nodes $s_i$ and $t_i$ for each clause $c_i$: the set $S$ includes $p_j^s$ and $n_j^s$ for each variable $x_j$, and $s_i$ for each clause $c_i$, while the set $T$ includes $p_j^t$ and $n_j^t$ for each variable $x_j$, and $t_i$ for each clause $c_i$.
- The set of arcs of $G$ includes the six arcs

$$(p_j^s, p_j), (p_j, p_j^t), (p_j, n_j), (n_j, p_j), (n_j^s, n_j), (n_j, n_j^t)$$

  for each variable $x_j$. It also includes the arc $(t_i, s_i)$ for each clause $c_i$.
- For each clause $c_i = l_i^1 \vee l_i^2 \vee l_i^3$, let $u_i^h = p_j$ and $v_i^h = n_j$(respectively, $u_i^h = n_j$ and $v_i^h = p_j$) if $l_i^h = x_j$ (respectively, $l_i^h = \neg x_j$) for $h = 1, 2, 3$. For any $h$ with $h = 1, 2, 3$, the set of arcs of $G$ includes the two arcs $(s_i, u_i^h)$ and $(v_i^h, t_i)$.

Let us prove that $\varphi$ is satisfiable if and only if $G$ includes a maximal DAG whose set of sources (respectively, sinks) is equal to $S$ (respectively, $T$).

- **Only if.** If $\varphi$ is satisfiable, let $\tau$ be a truth-assignment that satisfies $\varphi$. For each variable $x_j$ such that $\tau(x_j) = \texttt{true}$ (respectively, $\tau(x_j) = \texttt{false}$) we include in the FAS the arc $(n_j, p_j)$ (respectively, $(p_j, n_j)$). Moreover, for each clause $c_i$, we include in the FAS the arc $(t_i, s_i)$ (see Figure 5). Clearly, the resulting subgraph $S$ is a DAG whose set of sources (respectively, sinks) is equal to $S$ (respectively, $T$). Moreover, $S$ is maximal since no arc in the FAS can be removed by the FAS itself: otherwise, either a two-node variable cycle would be created or, for some clause $c_j$, at least one six-node cycle would be created corresponding to a true literal in $c_j$.
- **If.** If $S$ is a maximal DAG whose set of sources (respectively, sinks) is equal to $S$ (respectively, $T$), then, for each clause $c_i$, the arc $(t_i, s_i)$ is not in $S$. This implies that, since $S$ is maximal, for each variable $x_j$, exactly one between the two arcs $(p_j, n_j)$ and $(n_j, p_j)$ are included in $S$: all other arcs are included in $S$. Let $\tau$ be a truth-assignment defined as follows: for each variable $x_j$, $\tau(x_j) = \texttt{true}$ if and only if $(p_j, n_j)$ is in $S$. We now prove that this assignment

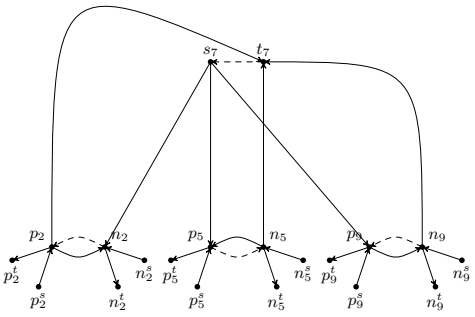


Fig. 5: The dyrected acyclic subgraph corresponding to the truth assignement $\tau(x_2) =$ `true`, $\tau(x_5) =$ `false`, and $\tau(x_9) =$ `true` that satisfies the clause $c_7 = \neg x_2 \vee x_5 \vee x_9$: the dashed arcs are in the FAS

satisfies $\varphi$. Suppose that there exists a clause $c_i = l_i^1 \vee l_i^2 \vee l_i^3$ which is not satisfied by $\tau$ (see Figure 6): this implies that the three cycles involving the arc $(t_i, s_i)$ are broken both by this arc and by the three arcs $(u_i^h, v_i^h)$ not in $S$, for $h = 1, 2, 3$. Hence, $S$ is not maximal since the arc $(t_i, s_i)$ can be added to $S$ without creating any new cycle. This contradicts the hypothesis on $S$.

Since a maximal DAG contained in a graph of all black nodes is a story, we have thus proved that the problem of finding a story with a specific set of sources *and* a specific set of sinks is NP-complete. □

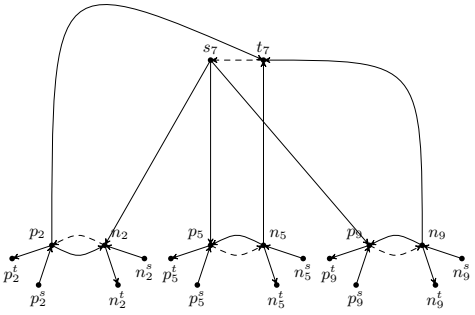


Fig. 6: A dyrected acyclic subgraph (the dashed arcs are in the FAS) corresponding to the truth assignement $\tau(x_2) =$ `true`, $\tau(x_5) =$ `false`, and $\tau(x_9) =$ `false` that does not satisfy the clause $c_7 = \neg x_2 \vee x_5 \vee x_9$: the DAG is not maximal since the arc $(t_7, s_7)$ can be taken out from the FAS.

It is easy to modify the previous reduction in order to prove that the same result holds even if we specify only the set of sources *or* only the set of sinks.

## 7 Alternative definition of a story

It is clear that, according to our definition of a story, no white node can be either source or target in the original graph, since otherwise such a white node would not belong to any story. This implies that the original graph can be seen as the union of a finite set $\mathcal{P}$ of paths between black nodes: in particular, if $\mathcal{P}$ includes all paths between black nodes, then it is easy to verify that a story is a subset $\mathcal{S}$ of $\mathcal{P}$ such that the graph induced by $\mathcal{S}$ is acyclic and there exists no path $p$ in $\mathcal{P} - \mathcal{S}$ for which $\mathcal{S} \cup \{p\}$ induces an acyclic graph.

A natural question is whether the problem changes when the set $\mathcal{P}$ is given as input along with the graph: the answer to this question is affirmative, as shown in Figure 7. However, we can now prove that stories (according to the new definition) cannot be enumerated in polynomial-incremental delay.

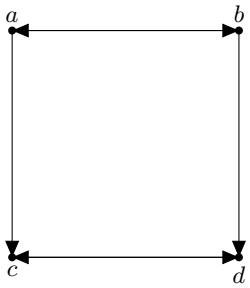


Fig. 7: Graph obtained by two paths $(a, b, d, c)$ and $(b, a, c, d)$. According to the alternative definition, this graph clearly contains only two stories, which correspond to the two paths. According to the original definition, instead, the graph contains the following four minimal SAS: $\{(a, b), (c, d)\}$, $\{(a, b), (d, c)\}$, $\{(b, a), (c, d)\}$, and $\{(b, a), (d, c)\}$. Note that these four minimal SAS originated four stories which are all different from the two stories obtained according to second definition.

**Theorem 11.** *Enumerating stories (according to the new definition) cannot be done in polynomial incremental delay.*

*Proof.* Let $\mathcal{C}$ be a collection of subsets of a domain set $X$. A *hitting set* for $\mathcal{C}$ is a subset $H$ of $X$ such that, for any $C \in \mathcal{C}$, $H \cap C \neq \emptyset$. We now reduce $\mathcal{C}$ to a collection $\mathcal{P}$ of paths, such that there is a bijective correspondence between hitting sets for $\mathcal{C}$ and stories for $\mathcal{P}$. Without loss of generality, we can assume that there exists an ordering of the elements of $X$ and of the elements of $\mathcal{C}$. For any subset $C_i \in \mathcal{C}$ such that $|C_i| = k_i$, let $C_i = \{x_i^1, \ldots, x_i^{k_i}\}$ with $x_i < x_{i+1}$.

For any element $x_j$ of $X$, let $o_j$ be the number of occurrences of $x_j$, let $s_j^h$ be the index of the element of $\mathcal{C}$ which contains the $h$-th occurrence of $x_j$, and let $p_j^h$ be the position of $x_j$ in $C_{s_j^h}$. For each $x_j$, $\mathcal{P}$ will contain the path $P_j$ that starts from the node corresponding to the first occurrence of $x_j$, follows the edge leaving this node, and then goes to the node corresponding to the next occurrence. This is repeated until the path reaches the node corresponding to the last occurrence: $P_j$ ends by following the edge leaving this latter node. Formally, this path is defined as follows:

$$P_j = (x_{s_j^1}^{p_j^1}, x_{s_j^1}^{\sigma(p_j^1)}), (x_{s_j^1}^{\sigma(p_j^1)}, x_{s_j^2}^{p_j^2}),$$
$$(x_{s_j^2}^{p_j^2}, x_{s_j^2}^{\sigma(p_j^2)}), (x_{s_j^2}^{\sigma(p_j^2)}, x_{s_j^3}^{p_j^3}), ..., (x_{s_j^{o_j}}^{p_j^{o_j}}, x_{s_j^{o_j}}^{\sigma(p_j^{o_j})}),$$

where

$$\sigma(p_j^h) = \begin{cases} p_j^h + 1 \text{ if } p_j^h < k_{s_j^h}, \\ 1 \qquad \text{otherwise.} \end{cases}$$

The graph induced by $\mathcal{P}$ contains all the edges $(x_i^j, x_i^{j+1})$ and the edge $(x_i^{k_i}, x_i^1)$ (that is, the graph contains a cycle between the nodes $x_i^1, \ldots, x_i^{k_i}$ corresponding to $C_i$). An example of the reduction is shown in Figure 8.

Let $H = \{x_{j_1}, \ldots, x_{j_h}\}$ be a hitting set for $\mathcal{C}$. Then, $\mathcal{F} = \bigcup_{u=1}^{h} \{P_{j_u}\}$ is a feedback path set. Indeed, the path $P_{j_u}$ breaks all cycles corresponding to the sets containing $x_{j_u}$: since $H$ is a hitting set, this implies that all cycles are broken. Moreover, if $H$ is minimal, then $\mathcal{F}$ is also minimal. Referring to the example of Figure 8, if $H$ is equal to $\{A, C\}$, then $\mathcal{F}$ contains the paths

$$(A_1, B_1), (B_1, A_2), (A_2, B_2), (B_2, A_3), (A_3, D_3)$$

and

$$(C_1, D_1), (D_1, C_2), (C_2, D_2).$$

On the contrary, if $\mathcal{F} = \{P_{j_1}, \ldots, P_{j_h}\}$ is a feedback path set, then $H = \{x_{j_1}, \ldots, x_{j_h}\}$ is a hitting set for $\mathcal{C}$. Indeed, since $\mathcal{P} - \mathcal{F}$ is a DAG, then all cycles are broken by at least one path in $\mathcal{F}$. This implies that for any set $C_i$, there exists at least one path in $\mathcal{F}$ which corresponds to an element in $C_i$. Hence, $H$ is a hitting set. Once again, if $\mathcal{F}$ is minimal, then $H$ is also minimal. Referring to the example of Figure 8, if $\mathcal{F}$ contains the paths

$$(B_1, C_1), (C_1, B_2), (B_2, E_2),$$

and

$$(D_1, A_1), (A_1, D_2), (D_2, E_1), (E_1, D_3), (D_3, F_1)$$

then $H$ is equal to $\{B, D\}$. Since enumerating all minimal hitting sets cannot be done in polynomial incremental delay [2], then the theorem follows. □

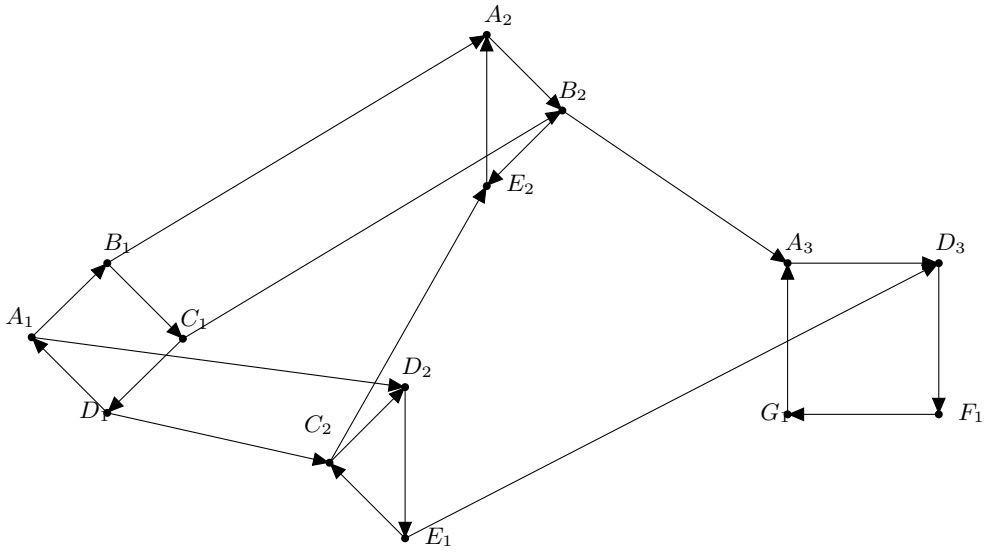


Fig. 8: An example of reduction: $C_1 = \{A, B, C, D\}$, $C_2 = \{C, D, E\}$, $C_3 = \{A, B, E\}$, and $C_4 = \{A, D, F, G\}$.

In this paper we have mainly focused our attention on the first definition of stories, since this definition seems to fit better with the informal subnetwork definition the biologists are looking for.

## 8 Conclusion

In this paper, we have introduced the new notion of story, which is a maximal acyclic subgraph of a directed graph in which only specified nodes can be sources or targets. We have proved some complexity results and designed some algorithms for enumerating all possible stories of a graph. The main question left open by the paper is to establish the complexity of the enumeration problem.

## References

1. M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.
2. V. Gurvich and L. Khachiyan. On generating the irredundant conjunctive and disjunctive normal forms of monotone boolean functions. *Discrete Applied Mathematics*, 96-97:363 – 373, 1999.
3. V. Lacroix, L. Cottret, P. Thbault, and M. F. Sagot. An Introduction to Metabolic Networks and Their Structural Analysis. *TCBB*, 5(4):594–617, 2008.

4. G. Madalinski, E. Godat, S. Alves, D. Lesage, E. Genin, P. Levi, J. Labarre, J.-C. Tabet, E. Ezan, and C. Junot. Direct introduction of biological samples into a ltq-orbitrap hybrid mass spectrometer as a tool for fast metabolome analysis. *Analytical Chemistry*, 80(9):3291–3303, 2008. PMID: 18351782.
5. B. Schwikowski and E. Speckenmeyer. On enumerating all minimal solutions of feedback problems. *Discrete Applied Mathematics*, 117(1-3):253 – 265, 2002.