The Complexity of Finding Effectors

Laurent Bulteau, Stefan Fafianie, Vincent Froese, Rolf Niedermeier, and Nimrod Talmon(✉)

Institut Für Softwaretechnik Und Theoretische Informatik,
TU Berlin, Berlin, Germany
{l.bulteau,nimrodtalmon77}@gmail.com,
{stefan.fafianie,vincent.froese,rolf.niedermeier}@tu-berlin.de

Abstract. The NP-hard Effectors problem on directed graphs is motivated by applications in network mining, particularly concerning the analysis of (random) information-propagation processes. In the corresponding model the arcs carry probabilities and there is a probabilistic diffusion process activating nodes by neighboring activated nodes with probabilities as specified by the arcs. The point is to explain a given network activation state best possible using a minimum number of "effector nodes"; these are selected before the activation process starts.

We complement and extend previous work from the data mining community by a more thorough computational complexity analysis of Effectors, identifying both tractable and intractable cases. To this end, we also exploit a parameterization measuring the "degree of randomness" (the number of 'really' probabilistic arcs) which might prove useful for analyzing other probabilistic network diffusion problems.

1 Introduction

To understand and master the dynamics of information propagation in networks (biological, chemical, computer, information, social) is a core research topic in data mining and related fields. A prominent problem in this context is the NP-hard problem Effectors [10]: The input is a directed (influence) graph with a subset of nodes marked as active (the target nodes) and each arc of the graph carries an influence probability between 0 and 1. The task is to find few "effector nodes" that can "best explain" the set of given active nodes, that is, the activation state of the graph; herein, in one round (this is known as the independent cascade model [9]) an activated node (initially consisting only the chosen effectors) can activate every out-neighbor with the corresponding arc probability; see Sect. 2 for a formal model and problem definition. It is important to

A full version is available at http://arxiv.org/abs/1411.7838.

Laurent Bulteau—Supported by the Alexander von Humboldt Foundation, Bonn, Germany.
Stefan Fafianie—Supported by the DFG Emmy Noether-program (KR 4286/1).
Vincent Froese—Supported by the DFG, project DAMM (NI 369/13).
Nimrod Talmon—Supported by DFG Research Training Group MDS (GRK 1408).

R. Jain et al. (Eds.): TAMC 2015, LNCS 9076, pp. 224–235, 2015.
DOI: 10.1007/978-3-319-17142-5_20

note that we allow effectors to be chosen from the *whole* set of graph nodes and not only from the set of target nodes. This makes our model, in a sense, more general than the original one by Lappas et al. [10].[1] Our main contribution is to extend and clarify research on the computational complexity status of Effectors, which has been initiated by Lappas et al. [10]. In particular, as *probabilistic* information propagation is central in this model as well as in other network diffusion models, we put particular emphasis on studying how the "degree of randomness" in the network governs the computational complexity. Moreover, compared to previous work, we make an effort to present the results in a more formal setting conducting a rigorous mathematical analysis.

Informally speaking (concrete statements of our results appear in Sect. 2 after having provided formal definitions), we gained the following main insights.

- With unlimited degree of randomness, finding effectors is computationally very hard. In fact, even computing the "cost" (how well does a set of effectors explain a given activation state) of a *given* set of effectors is intractable. This significantly differs from deterministic models.
- Even if the directed input graph is acyclic, then this does *not* lead to a significant decrease of the computational complexity.
- Bounding the degree of randomness (in other words, bounding the number of arcs with probability different from 0 or 1), that is, *parameterizing on the degree of randomness*, yields some encouraging (fixed-parameter) tractability results for otherwise intractable cases.
- We identify some flaws in the work of Lappas et al. [10] (see the last part of Sect. 4.3 for details), who claim one case to be intractable which in fact is tractable and one case the other way around.

Admittedly, in real-world applications (where influence probabilities are determined through observation and simulation, often involving noise) the number of probabilistic arcs may be high, thus rendering the parameter "number of probabilistic arcs" doubtful. However, note that finding effectors is computationally very hard (also in terms of polynomial-time approximability) and so in order to make the computation of a solution more feasible one might round up (to 1) arc probabilities which are close to 1 and round down (to 0) arc probabilities which are close to 0. Thus, we can achieve a tradeoff between running time and accuracy of the result. Depending on the degree of rounding (as much as a subsequent fixed-parameter algorithm exploiting the mentioned parameter would "allow") in this way one might at least find a good approximation of an optimal set of effectors in reasonable time.

Related Work. Our main point of reference is the work of Lappas et al. [10]. Indeed, we use a slightly different (and more general) problem definition: they define the effectors to be necessarily a subset of the target nodes, whereas we

[1] We conjecture that both models coincide if we have unlimited budget, that is, if the number of chosen effectors does not matter. On the contrary, they do not coincide if we have limited budget, see Sect. 2.

allow the effectors to form an arbitrary subset of the nodes. It turns out that these two definitions really yield different problems (see Sect. 2 for an extensive discussion of the two models). The also NP-hard special case where all nodes are target nodes (and hence where the two models above clearly coincide) is called Influence Maximization and is also well studied in the literature [4,6,9]. Finally, a closely related deterministic version (called Target Set Selection) with the additional difference of having node-individual thresholds specifying how many neighboring nodes need to be active to make a node active has also been extensively studied, in particular from a parameterized complexity point of view [3,5,11].

2 Preliminaries and Model Discussion

In this section, we first provide the formal framework, overview our results, and explain our modeling, particularly discussing the difference between our model and the one by Lappas et al. [10].

Preliminaries. We basically use the same definitions as Lappas et al. [10] except for few differences in notation.

Influence Graphs. An *influence graph* $G = (V, E, w)$ is a simple directed graph equipped with a function $w : E \to (0, 1] \cap \mathbb{Q}$ assigning an *influence weight* to each arc $(u \to v) \in E$ which represents the *influence of node u on node v*. We denote the number of nodes in G by $n := |V|$ and the number of arcs in G by $m := |E|$.

Information Propagation. We consider the following information-propagation process, called the *Independent Cascade (IC)* model [9]. Within this model, each node is in one of two states: *active* or *inactive*. When a node u becomes active for the first time, at time step t, it gets a single chance to activate its out-neighbors. Specifically, u succeeds in activating a neighbor v with probability $w(u \to v)$. If u succeeds, then v will become active at step $t+1$. Otherwise, u cannot make any more attempts to activate v in any subsequent round. As usual, we assume that the precision of the probabilities determined by the function w is polynomially bounded in the number n of nodes in the input graph.

Cost Function. For a given influence graph $G = (V, E, w)$, subset $X \subseteq V$ of effectors, and subset $A \subseteq V$ of active nodes, we define a cost function

$$C_A(G, X) := \sum_{v \in A} (1 - p(v|X)) + \sum_{v \in V \setminus A} p(v|X),$$

where for each $v \in V$, we define $p(v|X)$ to be the probability of v being active after the termination of the information-propagation process starting with X as the active nodes. An alternative definition is that $C_A(G, X) := \sum_{v \in V} C_A(v, X)$, where $C_A(v, X) := 1 - p(v|X)$ if $v \in A$ and $C_A(v, X) := p(v|X)$ if $v \notin A$.

Main Problem Definition. Our central problem Effectors is formulated as a decision problem—it relates to finding few nodes which best explain (lowest cost) the given network activation state specified by a subset $A \subseteq V$ of nodes.

Effectors
Input: An influence graph $G = (V, E, w)$, a set of target nodes $A \subseteq V$, a budget $b \in \mathbb{N}$, and a cost $c \in \mathbb{Q}$.
Question: Is there a subset $X \subseteq V$ of effectors such that $|X| \leq b$ and $C_A(G, X) \leq c$?

We will additionally consider the related problem Effectors-Cost (see Sect. 3) where the set X of effectors is already given and one has to determine its cost.

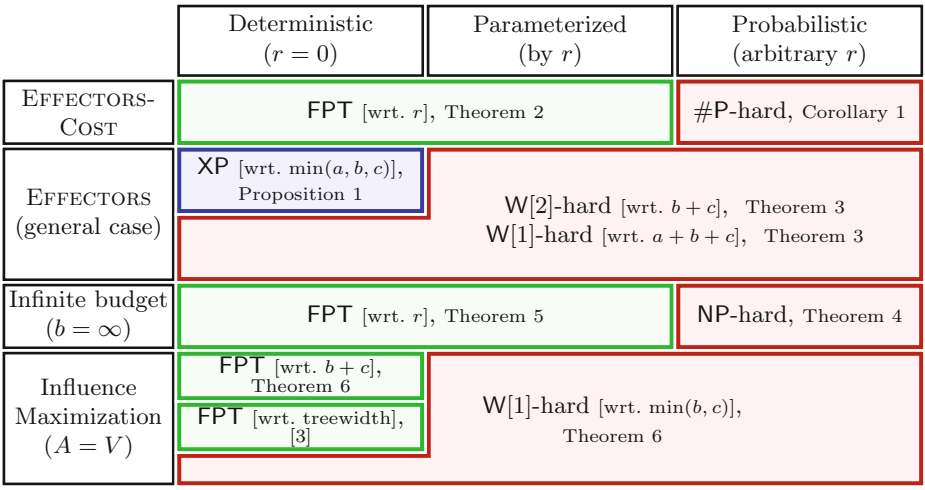
Parameters. The most natural parameters to consider for a parameterized complexity analysis are the maximum number b of effectors, the cost value c, and the number $a := |A|$ of target nodes. Moreover, we will be especially interested in quantifying the amount of randomness in the influence graph. To this end, consider an arc $(u \to v) \in E$: if $w(u \to v) = 1$, then this arc is not probabilistic. We define the parameter number r of probabilistic arcs, that is, $r := |\{(u \to v) \in E : 0 < w(u \to v) < 1\}|$. We will also briefly discuss the parameterization by the treewidth of the underlying undirected graph.

Graph Theory. We use the acronym DAG for directed acyclic graphs. The DAG of strongly connected components of a directed graph is called its *condensation*. A *directed tree* is an arbitrary orientation of an undirected tree.

Computational Complexity. We assume familiarity with the basic notions of algorithms and complexity. Several of our results will be cast using the framework of parameterized complexity analysis. An instance (I, k) of a parameterized problem consists of the actual instance I and an integer k being the *parameter* [7,8,12]. A parameterized problem is called *fixed-parameter tractable* (FPT) if there is an algorithm solving it in $f(k) \cdot |I|^{O(1)}$ time, whereas an algorithm with running time $O(|I|^{f(k)})$ only shows membership in the class XP (clearly, FPT $\subseteq$ XP). Thus, achieving fixed-parameter tractability is computationally much more attractive. One can show that a parameterized problem L is (presumably) not fixed-parameter tractable by devising a *parameterized reduction* from a W[1]-hard or W[2]-hard problem (such as Clique or Set Cover, respectively, each parameterized by the solution size) to L. A parameterized reduction from a parameterized problem L to another parameterized problem L' is a function that, given an instance (I, k), computes in $f(k) \cdot |I|^{O(1)}$ time an instance (I', k') (with $k' \leq g(k)$) such that $(I, k) \in L \Leftrightarrow (I', k') \in L'$. We will also consider counting problems of the form "compute func(x)". Informally speaking, we can associate a decision problem in NP to a counting problem in #P. Then, analogously to NP-hardness, showing that a counting problem is #P-hard gives strong evidence for the intractability of the counting problem.

Our Results. Before we discuss our model and the one by Lappas et al. [10], we overview our main results. We will treat the subproblem Effectors-Cost in Sect. 3, and Effectors in Sect. 4. Our results are summarized in Table 1. Note that most of our results transfer to the model of Lappas et al. [10]. In particular, this implies that their claims that the "zero-cost" special case is NP-hard [10, Lemma 1] and that the deterministic version is polynomial-time solvable are

Table 1. Computational complexity of the different variants of Effectors. Note that all hardness results hold also for DAGs. The parameter a stands for the number of active nodes, b for the budget, c for the cost value, and r for the number of probabilistic arcs.



both flawed, because from our results exactly the opposite follows (see the last part of Sect. 4.3 for details). Due to lack of space, most of the proofs are omitted. For full formal proofs, refer to the full version (available at http://arxiv.org/abs/1411.7838).

Model Discussion. Our definition of Effectors differs from the problem definition of Lappas et al. [10] in that we do not require the effectors to be chosen among the target nodes. Before pointing out possible advantages and motivating our problem definition, we give a simple example illustrating the difference between these two problems.

Consider the influence graph in Fig. 1, consisting of one non-target node (white) having three outgoing arcs of weight 1 to three target nodes (black). Clearly, for $b = c = 1$, this is a "no"-instance if we are only allowed to pick target nodes as effectors since the probability of being active will be 0 for two of the three target nodes in any case, which yields a cost of at least 2. According to our problem definition, however, we are allowed to select the non-target node, which only incurs a cost of 1, showing that this is a "yes"-instance. We think that our model captures the natural assumption that an effector node does not have to remain active forever. Indeed, the modeling of Lappas et al. [10] might be interpreted as a "monotone version" as for example discussed by Askalidis et al. [2], while in this sense our model allows for "non-monotone explanations".

Clearly, if all nodes are target nodes (this particular setting is called *Influence Maximization*), then the two models coincide. Furthermore, we strongly conjecture that if we have an unlimited budget, then it suffices to search for a solution among the target nodes, that is, also for $b = \infty$, we believe that the two problem definitions are equivalent.

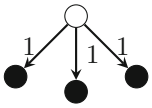


Fig. 1. A small example where it is optimal to choose a non-target node as effector (Color figure online).

Conjecture 1. For $b = \infty$, it holds that for every "yes"-instance (G, A, b, c) of Effectors there exists a solution $X \subseteq A$.

At least for directed trees (that is, the underlying undirected graph is a tree) we can prove Conjecture 1. The idea of proof is that if an optimal solution contains a non-target node, then this node only influences nodes reachable from it via paths that do not visit other nodes in the solution. Within this smaller tree of influenced nodes there must be some subtrees rooted at target nodes such that the expected cost for such a subtree is smaller if its target root node is activated during the propagation process compared to the case when it is not. Choosing these target nodes directly as effectors, replacing the non-target node, yields another optimal solution with fewer non-target nodes.

Theorem 1. *Conjecture 1 holds for directed trees.*

3 Computing the Cost Function

We consider the problem of computing the cost for a given set of effectors.

Effectors-Cost
Input: An influence graph $G = (V, E, w)$, a set of target nodes $A \subseteq V$, and a set of effectors $X \subseteq V$.
Compute: The cost $C_A(G, X)$.

Effectors-Cost is polynomial-time solvable on directed trees [10]. On the contrary, Effectors-Cost is unlikely to be polynomial-time solvable already on DAGs. This follows from a result by Wang et al. [13, Theorem 1]. They show that computing the expected number of activated nodes for a single given effector is #P-hard on DAGs. Note that for the case $A = \emptyset$, the cost equals the expected number of activated nodes at the end of the propagation process.

Corollary 1. Effectors-Cost on DAGs is #P-hard for $|A| = 0$ and $|X| = 1$.

On the positive side, Effectors-Cost is fixed-parameter tractable with respect to the number r of probabilistic arcs. The general idea is to recursively simulate the propagation process, branching over the probabilistic arcs, and to compute a weighted average of the final activation state of the graph.

Theorem 2. Effectors-Cost *can be solved in* $O(2^r \cdot n(n+m))$ *time, where* r *is the number of probabilistic arcs.*

4 Finding Effectors

We treat the general variant of Effectors in Sect. 4.1, the special case of unlimited budget in Sect. 4.2, and the special case of influence maximization in Sect. 4.3.

4.1 General Model

We consider the parameters number a of target nodes, the budget b, and the cost c. We first notice that if at least one of them equals zero, then Effectors is polynomial-time solvable. This holds trivially for parameters a and b; simply choose the empty set as a solution. This is optimal for $a = 0$, and the only feasible solution for $b = 0$. For parameter c, the following holds, using a simple decomposition into strongly connected components.

Lemma 1. For $c = 0$, Effectors can be solved in $O(n + m)$ time.

Based on Lemma 1, by basically checking all possibilities in a brute-force manner, we obtain simple polynomial-time algorithms for Effectors in the cases of a constant number a of target nodes, budget b, or cost c.

Proposition 1. *For $r = 0$,* Effectors *is in* XP *with respect to each of the parameters a, b, and c.*

In the following, we show that, even for $r = 0$, Effectors is W[1]-hard with respect to the *combined* parameter (a, b, c), and even W[2]-hard with respect to the *combined* parameter (b, c). We briefly sketch the proof of the first statement, and mention that the second statement is proven by a reduction from the W[2]-complete Dominating Set problem.

Theorem 3. *The following statements hold.*

1. Effectors*, parameterized by the combined parameter (a, b, c), is* W[1]*-hard, even if $r = 0$ and the influence graph is a DAG.*
2. Effectors*, parameterized by the combined parameter (b, c), is* W[2]*-hard, even if $r = 0$ and the influence graph is a DAG.*

Proof (Sketch for the first statement). We describe a parameterized reduction from the W[1]-hard problem Multi-Colored Clique, which asks for the existence of a colorful clique of size k in a simple and undirected graph whose vertices are colored with k colors. Given an instance of Multi-Colored Clique$(G = (V, E), k)$, we construct an instance of Effectors with $b = \binom{k}{2}$, $c = \binom{k}{2} + k$, and an influence graph defined as follows. Add $\binom{k}{2} + k + 1$ nodes for each pair of distinct colors. Let us call these nodes *color-pair nodes*. Now, add a *vertex node* n_v for each $v \in V$, add an *edge node* $e_{u,v}$ for each $e = \{u, v\} \in E$, and add arcs $\{e_{u,v} \to n_u, e_{u,v} \to n_v\}$. For each edge, let L be the color-pair nodes corresponding to the colors of u and v, and add arcs $\{e_{u,v} \to l \mid l \in L\}$. Finally, let the set of target nodes A contain all color-pair nodes and set the influence weights of all arcs to 1. □

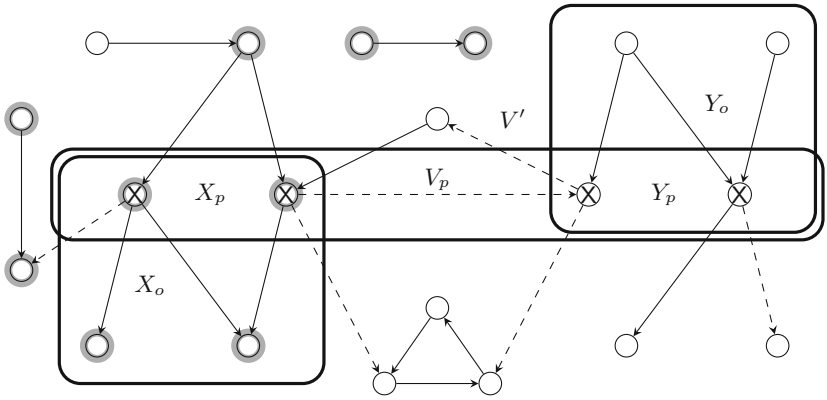


Fig. 2. Illustration for Theorem 5. Effectors of a solution are marked with an aura. Probabilistic arcs are dashed, and nodes of V_p (with an outgoing probabilistic arc) are marked with a cross. For readability, target nodes are not represented. Intuitively, the algorithm guesses the partition of V_p into X_p (effectors) and Y_p (non-effectors). Node set X_p (respectively, Y_p) is then extended to its closure X_o (respectively, its closure Y_o in the reverse graph). The remaining nodes form a deterministic subgraph $G[V']$, in which effectors, forming the set X', are selected by solving an instance of Maximum Weight Closure.

4.2 Special Case: Unlimited Budget

Here, we concentrate on a model variant where we are allowed to choose any number of effectors, that is, the goal is to minimize the overall cost with an unlimited budget of effectors. In general, Effectors with unlimited budget remains intractable, though (via reduction from a $\#P$-hard counting problem).

Theorem 4. *If* $\mathsf{P} \neq \mathsf{NP}$*, then* Effectors*, even with unlimited budget, is not polynomial-time solvable on DAGs.*

However, with unlimited budget, Effectors is fixed-parameter tractable with respect to the number r of probabilistic arcs.

Theorem 5. *If* $b = \infty$*, then* Effectors *is solvable in* $O(4^r \cdot n^4)$ *time, where* r *is the number of probabilistic arcs.*

Proof. The general idea is to fully determine the probabilistic aspects of the graph, and then to remove all of the corresponding nodes and arcs. We can show that this leaves an equivalent "deterministic graph" that we can solve using a reduction to the problem Maximum Weight Closure, which is itself polynomial-time solvable by a polynomial-time reduction to a flow maximization problem [1, Chapter 19].

Maximum Weight Closure
Input: A directed graph $G = (V, E)$ with weights on the vertices.
Compute: A maximum-weight set of vertices $X \subseteq V$ with no arcs going out of the set.

We start with some notation (see Fig. 2 for an illustration). For an input graph $G = (V, E)$, let $E_p := \{(u \to v) \in E \mid w(u \to v) < 1\}$ denote the set of probabilistic arcs and let $V_p := \{u \mid (u \to v) \in E_p\}$ denote the set of nodes with at least one outgoing probabilistic arc. For a node $v \in V$, let $\mathrm{cl}_{\mathrm{det}}(v)$ ($\mathrm{cl}^{-1}_{\mathrm{det}}(v)$) denote the set of all nodes u such that there exists at least one deterministic path from v to u (respectively, from u to v), where a deterministic path is a path containing only deterministic arcs. We extend the notation to subsets V' of V and write $\mathrm{cl}_{\mathrm{det}}(V') = \bigcup_{v \in V'} \mathrm{cl}_{\mathrm{det}}(v)$ and $\mathrm{cl}^{-1}_{\mathrm{det}}(V') = \bigcup_{v \in V'} \mathrm{cl}^{-1}_{\mathrm{det}}(v)$. We call a subset $V' \subseteq V$ of nodes *deterministically closed* if and only if $\mathrm{cl}_{\mathrm{det}}(V') = V'$, that is, there are no outgoing deterministic arcs from V' to $V \setminus V'$.

Our algorithm will be based on a closer analysis of the structure of an optimal solution. To this end, let $G = (V, E, w)$ be an input graph with a set $A \subseteq V$ of target nodes and let $X \subseteq V$ be an optimal solution with minimum cost $C_A(G, X)$. Clearly, we can assume that X is deterministically closed, that is, $\mathrm{cl}_{\mathrm{det}}(X) = X$, since we have an infinite budget $b = \infty$.

We write V_p as a disjoint union of $X_p := V_p \cap X$ and $Y_p := V_p \setminus X$. We also use $X_o := \mathrm{cl}_{\mathrm{det}}(X_p)$, $Y_o := \mathrm{cl}^{-1}_{\mathrm{det}}(Y_p)$ and $V_o = X_o \cup Y_o$. Since X is deterministically closed, we have that $X_o \subseteq X$ and $Y_o \cap X = \emptyset$. We write $V' := V \setminus V_o$ and $X' := X \setminus X_o = X \cap V'$. Note that X' is deterministically closed in $G[V']$ and that $G[V']$ contains only deterministic arcs. Moreover, note that the sets X_o, Y_p, Y_o, V_o, and V', are directly deduced from the choice of X_p, and that for a given X_p, the set X' can be any closed subset of V'.

We first show that the nodes in V_o are only influenced by effectors in X_o, that is, for any node $v \in V_o$, it holds that $p(v|X) = p(v|X_o)$. This is clear for $v \in X_o$, since in this case $p(v|X) = p(v|X_o) = 1$. Assume now that there is a node $x \in X'$ with a directed path to $v \in Y_o$ that does not contain any node from X_o. Two cases are possible, depending on whether this path is deterministic. If it is, then since $v \in \mathrm{cl}^{-1}_{\mathrm{det}}(Y_p)$, then there exists a deterministic path from x to some $u \in Y_p$, via v. Hence, $x \in \mathrm{cl}^{-1}_{\mathrm{det}}(Y_p) = Y_o$, yielding a contradiction. Assume now that the path from x to v has a probabilistic arc and write $u \to u'$ for the first such arc. Hence, $x \in \mathrm{cl}^{-1}_{\mathrm{det}}(u)$ and $u \in V_p$. Since we assumed that the path does not contain any node from X_o, we have $u \notin X_p$, and therefore $u \in Y_p$. Again, we have $x \in \mathrm{cl}^{-1}_{\mathrm{det}}(Y_p)$, yielding a contradiction.

Hence, the nodes in V_o are not influenced by the nodes in X'. Now consider nodes in V'. Note that we have $p(v|X) = 1$ for $v \in X'$ and $p(v|X) = p(v|X_o)$ for $v \in V' \setminus X'$, since $G[V']$ is deterministic and X' is deterministically closed.

Overall, $C_A(v, X) = C_A(v, X_o)$ for all $v \in V \setminus X'$. The total cost of solution X can now be written as

$$\begin{aligned} C_A(G, X) &= \sum_{v \in V \setminus X'} C_A(v, X_o) + \sum_{v \in X'} C_A(v, X) \\ &= \sum_{v \in V} C_A(v, X_o) - \sum_{v \in X'} \left(C_A(v, X_o) - C_A(v, X)\right) \\ &= \alpha(X_o) - \beta(X_o, X'), \end{aligned}$$

where

$$\alpha(X_o) := \sum_{v \in V} C_A(v, X_o) \quad \text{and} \quad \beta(X_o, X') := \sum_{v \in X'} (C_A(v, X_o) - C_A(v, X)).$$

We further define, for all $v \in V'$, $\gamma(v, X_o) := 1 - p(v|X_o)$ if $v \in A$, and $\gamma(v, X_o) = p(v|X_o) - 1$ if $v \notin A$. Note that, for $v \in X'$, the difference $C_A(v, X_o) - C_A(v, X)$ is exactly $\gamma(v, X_o)$, hence $\beta(X_o, X') := \sum_{v \in X'} \gamma(v, X_o)$.

The algorithm can now be described directly based on the above formulas. Specifically, we branch over all subsets $X_p \subseteq V_p$ (note that the number of these subsets is upper-bounded by 2^r). For each such subset $X_p \subseteq V_p$, we can compute X_o and Y_o in linear time because this involves propagation only through deterministic arcs (outgoing for X_o and ingoing for Y_o). Then, for each node $v \in V$, we compute $p(v|X_o)$ using Theorem 2 in $O(2^r \cdot n(n+m))$ time. This yields the values $\alpha(X_o)$ and $\gamma(v, X_o)$ for each $v \in V'$. The closed subset X' of V' maximizing $\beta(X_o, X')$ is then computed as the solution of Maximum Weight Closure on $G[V']$ (which is solved by a maximum flow computation in $O(n^3)$ time), where the weight of any $v \in V'$ is $\gamma(v, X_o)$. Finally, we return the set $X_o \cup X'$ that yields the minimum value for $\alpha(X_o) - \beta(X_o, X')$. □

4.3 Special Case: Influence Maximization

In this section, we consider the special case of Effectors, called Influence Maximization, where all nodes are targets ($A = V$). Note that in this case the variant with unlimited budget and the parameterization by the number of target nodes are irrelevant.

In the influence maximization case, on deterministic instances, one should intuitively choose effectors among the "sources" of the influence graph, that is, nodes without incoming arcs (or among strongly connected components without incoming arcs). Moreover, the budget b bounds the number of sources that can be selected, and the cost c bounds the number of sources that can be left out. In the following theorem, we prove that deterministic Effectors remains intractable even if either one of these parameters is small, but, on the contrary, having $b + c$ as a parameter yields fixed-parameter tractability in the deterministic case. We mention that the first statement is proven by a reduction from the W[2]-hard Set Cover problem, while the second statement is proven by a reduction from the W[1]-hard Independent Set problem.

Theorem 6. *The following holds.*

1. Effectors, *parameterized by the maximum number b of effectors, is* W[2]*-hard, even if G is a deterministic ($r = 0$) DAG and all nodes are target nodes ($A = V$).*
2. Effectors, *parameterized by the cost c, is* W[1]*-hard, even if G is a deterministic ($r = 0$) DAG and all nodes are target nodes ($A = V$).*
3. *If $r = 0$ and $A = V$, then* Effectors *can be solved in $O(\binom{b+c}{b} \cdot (n+m))$ time.*

Treewidth as a Parameter. As Effectors is in general not polynomial-time solvable (unless P = NP (Theorem 4)), but polynomial-time solvable on trees, it is natural to consider the treewidth of the underlying undirected graph as a parameter. Indeed, treewidth is a well-known concept in algorithmic graph theory. Informally, treewidth measures how "tree-like" a graph is—trees have treewidth one. We note that for deterministic influence graphs ($r = 0$) under the influence maximization model ($A = V$), Effectors corresponds to a special case of a related problem, namely Target Set Selection (with constant thresholds), for which fixed-parameter tractability for the parameter treewidth is already known [3]. It is basically straightforward—but tedious and technical—to extend this algorithm to the case where some nodes are non-targets ($A \subsetneq V$). We conjecture that, for influence graphs with $r > 0$ probabilistic arcs, the problem is still fixed-parameter tractable for the combined parameter treewidth and r. The most challenging open question is whether Effectors is fixed-parameter tractable when parameterized by the treewidth, even with an unbounded number of probabilistic arcs.

Results in Contradiction with Lappas et al. [10]. The following two claims from the literature are contradicted by the results presented in this paper.

According to Lappas et al. [10, Lemma 1], in the Influence Maximization case with $c = 0$, Effectors is NP-complete. The reduction is incorrect: it uses a target node ℓ which influences all other vertices with probability 1 (in at most two steps). It suffices to select ℓ as an effector in order to activate all vertices, so such instances always have a trivial solution ($X = \{\ell\}$), and the reduction collapses. On the contrary, we prove in Lemma 1 that all instances with $c = 0$ can be solved in linear time.

According to the discussion of Lappas et al. [10] following their Corollary 1, there exists a polynomial-time algorithm for Effectors with deterministic instances (with $r = 0$). Note that the selection model corresponds to our own model in the case of Influence Maximization. However, the given algorithm is flawed: it does not consider the influence *between* different strongly connected components. Indeed, as we prove in Theorem 6, finding effectors under the deterministic model is NP-hard, even in the case of Influence Maximization.

5 Conclusion

We leave several challenges for future research. First, it remains to (dis)prove that Conjecture 1 also holds for arbitrary directed graphs. Further, we have made some unproven claims about (fixed-parameter) tractability when restricting Effectors to directed graphs whose underlying graphs have bounded treewidth. Two more general directions could be to extend our results concerning the parameter "degree of randomness" to other probabilistic diffusion models or to make the considered probabilistic information-propagation problems more tractable by developing simpler (and better to analyze) "linearized models"—the non-linearity in computing the activation probabilities of nodes appears to be an important cause for computational hardness.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Upper Saddle River (1993)
2. Askalidis, G., Berry, R.A., Subramanian, V.G.: Explaining snapshots of network diffusions: structural and hardness results. In: Cai, Z., Zelikovsky, A., Bourgeois, A. (eds.) COCOON 2014. LNCS, vol. 8591, pp. 616–625. Springer, Heidelberg (2014)
3. Ben-Zwi, O., Hermelin, D., Lokshtanov, D., Newman, I.: Treewidth governs the complexity of target set selection. Discrete Optim. **8**(1), 87–96 (2011)
4. Bharathi, S., Kempe, D., Salek, M.: Competitive influence maximization in social networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 306–311. Springer, Heidelberg (2007)
5. Chopin, M., Nichterlein, A., Niedermeier, R., Weller, M.: Constant thresholds can make target set selection tractable. Theory Comput. Syst. **55**(1), 61–83 (2014)
6. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66. ACM (2001)
7. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Springer, London (2013)
8. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Heidelberg (2006)
9. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM (2003)
10. Lappas, T., Terzi, E., Gunopulos, D., Mannila, H.: Finding effectors in social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1059–1068. ACM (2010)
11. Nichterlein, A., Niedermeier, R., Uhlmann, J., Weller, M.: On tractable cases of target set selection. Soc. Netw. Anal. Min. **3**(2), 233–256 (2013)
12. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, New York (2006)
13. Wang, C., Chen, W., Wang, Y.: Scalable influence maximization for independent cascade model in large-scale social networks. Data Min. Knowl. Disc. **25**(3), 545–576 (2012)