NETWORKS AND HETEROGENEOUS MEDIA
©American Institute of Mathematical Sciences
Volume **10**, Number **1**, March **2015**

doi:10.3934/nhm.2015.10.195

pp. **195–208**

EFFICIENT ALGORITHMS FOR ESTIMATING LOSS OF INFORMATION IN A COMPLEX NETWORK: APPLICATIONS TO INTENTIONAL RISK ANALYSIS

Santiago Moral

IT Risk, Fraud and Security
Research and Innovation for IT Risk, Fraud and Security
BBVA, Madrid, Spain
and
Centro de Investigación para la Gestión Tecnológica del Riesgo
BBVA - Universidad Rey Juan Carlos
Madrid, Spain

Victor Chapela

Suggestic Inc
Palo Alto, San Francisco Bay Area, California, USA

Regino Criado, Ángel Pérez and Miguel Romance

Departamento de Matemática Aplicada
Ciencia e Ingeniería de los Materiales y Tecnología Electrónica
Universidad Rey Juan Carlos
28933 Móstoles (Madrid), Spain

Abstract. In this work we propose a model for the diffusion of information in a complex network. The main assumption of the model is that the information is initially located at certain nodes and then is disseminated, with occasional losses when traversing the edges, to the rest of the network. We present two efficient algorithms, which we called max-path and sum-path, to compute, respectively, lower and upper bounds for the amount of information received at each node. Finally we provide an application of these algorithms to intentional risk analysis.

1. **Introduction and notation.** Information is power. The exchange of information plays a central role in science, technology and society and modelling this information's flow is one of the challenges of the scientific community. In our global-scale world, information moves through technological and social complex networks and comes to us as a result of dynamical processes on these complex networks that surround us. In fact, recent years have witnessed a growing interest in understanding the fundamental principles of how information, epidemic and ideas spread over large networks. There is a large scientific literature that analyses the diffusion of information on complex networks (see, for example [2, 1, 3, 4, 8, 9, 10, 11] and the references therein), including many dynamical models, the relationship between the structural properties and the diffusion processes, and many others. In this paper

2010 *Mathematics Subject Classification.* Primary: 05C90, 05C75; Secondary: 68M10, 94C15, 90B18.

Key words and phrases. Information diffusion, intentional complex network, intentional technological risk.

we present a model for information diffusion in a complex communication network in such a way that some amount of initial information is located at one or several information source nodes and then it travels to rest of the nodes through the links of the network. We will focus on estimating the amount of information that reaches each node of a network.

We start with a communication complex network, which is a directed graph $G = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is a (finite) set of nodes that represent the actors that transmit the information and $E \subseteq V \times V$ is the set of links that represent the interactions or interconnections that enables the information transmission between nodes. Some examples of this type of networks include social networks and computer networks [3, 4, 10], among others. In a communication network G, each node can be a producer or transmitter of information, and the diffusion will only take place by using the (topological) structure of G. We fix some nodes (one or more) that will be the producers of information, and we will call them *sources of information* and we consider the following dynamic on G:

1. In the beginning, the information it is only available for the nodes that are sources of information.
2. These information is propagated following the links of the networks, starting from the sources and disseminating to the rest of the nodes.
3. During the propagation process, the information is not transformed neither by the participating nodes or by the used links. That is, the information do not change or increase during all the process.
4. When the information is distributed from one node v_i to another node v_j by using a link $\ell = (v_i, v_j)$, only a fraction of the information contained in v_i is transmitted to node v_j. That is, if node v_i has an amount of information m_i, and the information is distributed from one node v_i to other node v_j by using the link $\ell = (v_i, v_j)$, the amount of information received in v_j is $\psi(\ell) \cdot m_i$, where $\psi(\ell) \in (0, 1]$ is the *information loss* of link ℓ and therefore we can consider that the original network $G = (V, E)$ is a directed weighted network, where the weight of each link is its information loss.
5. Since a node can have several in-links the total information received by a node is the *union* of the information received by each link. We say the *union* since information coming from different links can be coincident as we will see in the example presented in Figure 1.
6. This process is repeated until all the possible information is transmitted and the information received by each node stabilises.

This process models the dissemination of information in a communication network from sources of information to the rest of the nodes. For example, if G is a computer network that contains a (valuable) database in a node, the previous process quantifies the fraction of the database that is reachable from each node of the network when only a portion of the information of a node v_i is available to node v_j when connection (v_i, v_j) occurs.

Note that this process is related with the classic *maximal flow problem* in graphs introduced in [7], but in this case, the links has a relative capacity instead of an absolute capacity. That is, while in the classic *maximal flow problem* each node has associated a number that represents its maximal capacity (considered as the maximal amount of information that this link can transmit), in the previous model each link has associated a number that measures the fraction of information that is transmitted in each link. While the classic maximal flow problem is usually related to diffusion with a (physical) limitation of the bandwidth of each link, the previous process is related with a diffusion process with relative limitation in bandwidth of links, probably due to security or confidence reasons.

Let us start by giving a very simple example that illustrate this process. Consider the network $G = (V, E)$ with a set of five nodes $V = \{v_1, v_2, v_3, v_4, v_5\}$ presented in Figure 1. The *information loss* of each link is the number presented in each link. If we have a unique source of information at node v_1 with an amount of information $c \in (0, +\infty)$, then the amount of information transmitted to node v_2, v_3 and v_4 is $c/2$ because there is only one path that connects node v_1 with these nodes. On the other hand, the information received by node v_5 is again $c/2$, despite the fact that there are two different paths linking v_1 with v_5. This is due to the fact that the information coming from each path is the same, since it has a common part (link (v_1, v_2)) and therefore the information available at node v_5 is exactly the same information available at node v_2.

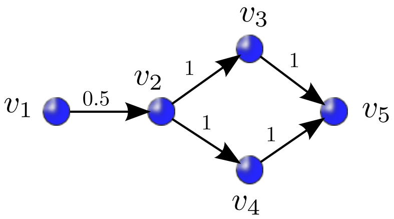

FIGURE 1. An example with 5 nodes. Source node is v_1. Numbers next to the links represent the proportion of information which is transmitted through that link.

Note that it is sometimes difficult to give the exact value of the amount of information transmitted to a node, since we only know the fraction of information distributed and not which part of the information it is actually transmitted. For example if we consider the network $G = (V, E)$ with a set of four nodes $V = \{v_1, v_2, v_3, v_4\}$ presented in Figure 2 and we take a unique source of information at node v_1 with an amount of information $c \in (0, +\infty)$, it is not clear the exact value of the amount of information transmitted to node v_4. Since there are two different paths that link v_1 with v_4, and from each of them the amount of information transmitted is $c/2$, we could expect that the amount of information available at node v_4 could be c, but we don't know if the information transmitted through the path $(v_1, v_2), (v_2, v_4)$ is disjoint or it coincides with the information sent through the path $(v_1, v_3), (v_3, v_4)$. As two extreme cases, we have the following scenarios: *(i)* on the one hand, if the information transmitted through the two paths are completely different (i.e. they are complementary), then the amount of information received by v_4 would be c, *(ii)* on the other hand, if the information sent through each of the two paths is the same, then the amount of information received by v_4 would be $c/2$.

There is no way of distinguishing the actual information transmitted though each link, since the presented process only takes into account the amount of information sent. In order to avoid this problem, we should consider a different (and more complicated) agent-based model that looks for the exact piece of information transmitted through every link. These new processes have their own disadvantages since they are more complex (in space) and in many cases it is not possible to

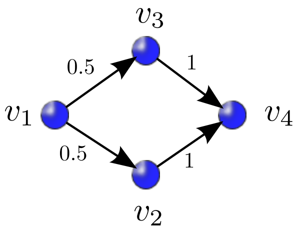

FIGURE 2. An example with 4 nodes. Source node is v_1. Numbers next to the links represent the proportion of information which is transmitted through that link.

take into account the actual information transmitted. As an alternative, in the previous model we could only give estimates of the information available at each node. For example, going back to the network presented in Figure 2, the amount of information available at node v_4 is between $c/2$ and c.

The main goal of this paper is presenting some efficient algorithms that compute good estimates of the amount of information transmitted to each node of a communication network. In particular in Section 2, we will present an algorithm (the *max-path algorithm*) that gives a lower bound of the amount of information available at each node and in Section 3 we will introduce an algorithm (the *sum-path algorithm*) for an upper bound of the amount of information. After illustrating the use of the algorithms in Section 4, we will present some applications to the computation of value in the Intentional Risk Analysis in Section 5. This is a subject which has recently attracted considerable attention for large corporations as well as for the scientific community, since risk calculation involves subjective parameters (*anonimity*, *accesibility*, *value*) and real incidents that are occurring on its own environment, beyond the basis of traditional risk assessment models: frequency and impact.

Next we introduce some definitions and notation which will be used later on. From a schematic point of view, a (directed) complex network is a mathematical object $G = (V, E)$ composed by a set of nodes or vertices $V = \{v_1 \ldots, v_n\}$ that are pairwise joined by oriented links or edges $E = \{\ell_1, \ldots, \ell_m\}$.

Now we will describe how the information will be spread throughout the network using a matrix. Given a directed graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$, and a map $\psi : E \longrightarrow (0, 1] \subseteq \mathbb{R}$, we call *information loss matrix* of G to the $n \times n$ matrix B, which has as inputs

$$B(i,j) = \begin{cases} \psi((v_i, v_j)) & \text{if} \quad (v_i, v_j) \in E \\ 0 \quad \text{if} & (v_i, v_j) \notin E. \end{cases}$$

Intuitively, the map ψ, which we will call *information loss map*, assigns to each edge the proportion of information which is preserved (i.e. one minus the proportion of information which is lost) when the information traverses that edge.

2. The max-path algorithm.

Let $\rho = (\lambda_1, \ldots, \lambda_m)$ a directed path in the graph $G = (V, E)$ with starting and end nodes $v_i, v_j \in V$ respectively. Assume that the starting node v_i has an associated information amount $M \in \mathbb{R}^+$ and that an information loss map $\psi : E \longrightarrow (0, 1]$ has been fixed. We define the *information* *transmitted through* ρ as the following quantity:

$$I(\rho) = M \cdot \prod_{k=1}^{m} \psi(\lambda_k).$$

We also define the *maximum information dispersed from* v_i *to* v_j as

$$MI_i(j) = \max_{\rho}\{I(\rho) : \rho \text{ is a directed path from } v_i \text{ to } v_j\}$$

for every $j \neq i$. For completeness we define $MI_i(i) := M$.

Next we provide an efficient algorithm, which we call *max-path*, for computing the value $MI_i(j)$ for every node $v_j \in V$. That is, our algorithm will compute, for every node v_j, the maximum amount of information transmitted from v_i to v_j by using every possible directed path which connects these two nodes. We emphasize that the algorithm does not need to explicitly compute all these paths in order to get the desired output.

In the sequel, given a vector $x \in \mathbb{R}^n$, we denote by $x[j]$ the j-th coordinate of x. Given a square matrix A we denote by A^T the transpose of the matrix A. The table named Algorithm 1 provides a description of max-path.

Note that $MP_i(v_j) = MI_i(j)$ for every $j \in \{1, \ldots, n\}$ and it is also remarkable that a tighter stop condition is presented. This condition is better than a trivial condition based on properties of the metric structure of the network (such as the diameter). Since the system is finite and the diffusion process is bounded and cumulative, the algorithm finishes in a finite number of steps. A naive stop condition is related with the diameter of the network from the source of information, since the diffusion process is based on the propagation of information along paths from the source. A more tighter stop condition proposed for the algorithm is the following:

“when $x_{k-1} = x_k$, stop the algorithm”

Next we will proof this last claim. We will denote by $len(\rho)$ the length of the directed path ρ and by

$$MI_{i,j,k} = \max_{\rho}\{I(\rho) : \rho \text{ is a directed path from } v_i \text{ to } v_j \text{ with } len(\rho) \leq k\}.$$

Lemma 2.1. *For every* $j \in \{1, \cdots, n\}$ *and* k *positive integer, we have* $x_k[j] = MI_{i,j,k}$.

Proof. First note that, for every step ℓ, we have $y_\ell[j] = \max_\rho\{I(\rho)\}$, taken from the directed paths ρ from v_i to v_j of length ℓ. This is obvious from the construction of the algorithm.

Note also that, again by the construction of the algorithm,

$$x_k[j] = \max_{\ell}\{y_\ell[j] : \ell \leq k\}.$$

Let ρ_0 the path such that $MI_{i,j,k}$ is attained at ρ_0 and let $\ell_0 = len(\rho_0)$. Then we have

$$x_k[j] = \max_{\ell}\{y_\ell[j] : \ell \leq k\} = y_{\ell_0}[j] = I(\rho_0) = MI_{i,j,k}.$$

□

Theorem 2.2. *If* k *is a positive integer such that* $x_{k-1} = x_k$ *then* $x_m = x_k$ *for every* $m > k$.

```
Algorithm 1: Max-path algorithm
  Input :
    • A directed network $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$.
    • An information loss map $\psi : E \longrightarrow (0, 1] \subseteq \mathbb{R}$.
    • An initial node $v_i \in V$.
    • An initial information amount $M \in \mathbb{R}^+$.
    • A stop condition and/or a maximum number $Max$ of iterations.
  Output: A map $MP_i : V \longrightarrow \mathbb{R}$ of the dispersed information from the node $v_i$.
  begin
      Step 0 (information loss matrix). From the map $\psi$ we construct the
      information loss matrix $B$ as described in section 1.
      Step 1 (initial vectors). We consider an initial vector $x_0 \in \mathbb{R}^n$ with value
      $M$ only in the position that corresponds to the node $v_i$, that is
          $x_0[j] = \left\{ \begin{array}{ccc} M & \text{if} & j = i \\ 0 & \text{if} & j \neq i. \end{array} \right.$
      We define as well $y_0 = x_0$.
      Step 2 (k-iteration). Given the vectors $x_{k-1}, y_{k-1} \in \mathbb{R}^n$ we build the
      vectors $x_k, y_k$ as follows:
          1. Given the vector $y_{k-1}$, we decompose it as a sum of vectors that have
             a single coordinate different from zero:
                 $y_{k-1} = \sum_{y_{k-1}[\ell] \neq 0} y_{k-1}[\ell] \cdot e_\ell := \sum_{y_{k-1}[\ell] \neq 0} u_{k-1}^{(\ell)},$
             where $e_\ell$ is the $\ell$-th vector of the canonical basis.
          2. We multiply each of those vectors by the matrix $B^T$:
                 $u_k^{(\ell)} = B^T \cdot u_{k-1}^{(\ell)}.$
          3. We settle $y_k$ as the maximum, coordinate by coordinate, of the results:
                 $y_k[j] = \max_\ell \{u_k^{(\ell)}[j]\}$
             for each $j = 1, \ldots, n$.
          4. In order to build $x_k$, we take maxima in every coordinate:
                 $x_k[j] = \max\{x_{k-1}[j], y_k[j]\}$
             for each $j = 1, \ldots, n$.
      Step 3 (output building). When the maximum number of iterations $Max$
      is reached, or the stop condition is fulfilled, no more iterations will be
      carried out. We denote by $x_K$ the last obtained vector. We set
          $MP_i(v_j) = x_K[j]$
      for every $j = 1, \ldots, n$.
  end
```

Proof. Let us reason by contradiction and assume that $x_{k-1} = x_k \neq x_{k+1}$. As $x_k[j] \leq x_{k+1}[j]$ for every j, there exists $j \in \{1, \cdots, n\}$ such that $x_k[j] < x_{k+1}[j]$. By Lemma 2.1, there exists a path ρ from v_i to v_j of length $k+1$ such that $I(\rho) > MI_{i,j,k}$.

Let $\rho = (\lambda_1, \ldots, \lambda_{k+1})$ and let $\lambda_{k+1} = (v_s, v_j)$. Consider $\rho_1 = (\lambda_1, \ldots, \lambda_k)$, which is a path from v_i to v_s of length k. Therefore there exists a path ρ_2 from v_i to v_s and $len(\rho_2) \leq k$ such that $I(\rho_2) = MI_{i,s,k} \geq I(\rho_1)$.

As $x_k = x_{k-1}$, again by Lemma 2.1, there exists a path ρ_3 from v_i to v_s and $len(\rho_3) \leq k-1$ such that $I(\rho_3) = I(\rho_2)$.

Let $\rho_3 = (\alpha_1, \ldots, \alpha_r)$ with $r \leq k-1$ and let $\rho_4 = (\alpha_1, \ldots, \alpha_r, \lambda_{k+1})$, which is a path from v_i to v_j with $len(\rho_4) \leq k$. Putting everything together, we have

$$I(\rho) = I(\rho_1) \cdot \psi(\lambda_{k+1}) \leq I(\rho_2) \cdot \psi(\lambda_{k+1}) = I(\rho_3) \cdot \psi(\lambda_{k+1}) = I(\rho_4).$$

This contradicts the fact that $I(\rho) > MI_{i,j,k}$ and ends the proof. □

As a consequence of previous Theorem, we get the correctness of the computation made by the algorithm in next Corollary:

Corollary 2.3. *If k is a positive integer such that $x_{k-1} = x_k$ then $x_k[j] = MI_i(j)$ for every $j \in \{1, \cdots, n\}$*

Proof. As $x_k[j] \leq x_{k+1}[j] \leq MI_i(j)$, Lemma 2.1 and Theorem 2.2 immediately imply the result. □

Remark 2.4. Consider $\mu_i = \max_\rho\{len(\rho)\}$ where ρ is a directed path with starting vertex v_i and no repeated vertices. From the previous results, it follows that after μ_i+1 steps, the output of the max-path algorithm will stabilize, that is, $x_{\mu_i} = x_{\mu_i+1}$. This is because paths with repeated edges will not produce an increment of the information amount at each node, as the weights of the edges are upper bounded by 1.

2.1. **On the complexity of max-path.** First note that in step 2.2 of the algorithm, the calculation of the vector $u_k^{(\ell)}$ may be done by multiplying the number $y_{k-1}[\ell]$ by the ℓ-th column of the matrix B^T. In this way, there is no need to make a explicit decomposition in step 2.1, it is enough to compute as many vectors $u_k^{(\ell)}$ as coordinates different from zero that the vector y_{k-1} has. Therefore, in step 2.2, n scalar times vector multiplications are performed at most, that is, at most n^2 real number multiplications.

In each of steps 2.3 and 2.4 the maximum between two real numbers is computed n times. As the number of iterations equals $\mu_i + 1$, which is upper bounded by n (in the worst-case scenario of a linear graph), we have the following upper bounds for the number of operations:

- Number of products of 2 real numbers $\leq n^3$
- Number of comparisons between 2 real numbers $\leq 2n^2$

3. An upper bound for the information transmission.

The max-path algorithm we presented in previous section, with a suitable stop condition, computes $MP_i(j)$ for every node v_j, that is, the minimum amount of information which arrives to v_j from v_i in our setting. However it is possible that, in some cases, more information than $MP_i(j)$ arrives to v_j. For example, assume that M is the initial amount of information at v_i, that $I(\rho) \leq M/2$ for every directed path in the graph and that there exist two paths, ρ_1 and ρ_2, from v_i to v_j, with no other common vertices, such that $I(\rho_1) = I(\rho_2) = M/2$. In this case, $MP_i(j) = M/2$. Nevertheless, it is conceivable that the half part of the information which travels through ρ_1 is disjoint which the half part which travels trough ρ_2, yielding a total amount of information M at v_j.

Therefore it is interesting to provide an upper bound for the amount of information which ends up in a node after diffusion. An obvious one is M, the initial information at v_i. But we can do better. Let us modify the max-path algorithm from Section 2 in the following way:

- In step 2.3, compute $y_k[j]$ as
$$y_k[j] := \min\{M, \sum_{\ell} u_k^{(\ell)}[j]\}.$$
- In step 2.4, compute $x_k[j]$ as
$$x_k[j] := \min\{M, x_{k-1}[j] + y_k[j]\}.$$

Let us give the name *sum-path* to the resulting algorithm. The intuition behind this new algorithm is that every time new information arrives to a node at certain step, it is added to the amount of information already stored at the node in the previous steps (but never going beyond the initial amount of information M). This corresponds to the optimistic assumption that "fresh" or "new" information arrives to the node.

Remark 3.1. If we run the sum-path algorithm with a fixed number of iterations equal to $\mu_i + 1$ (from Remark 2.4) we trivially get an upper bound for the total amount of information that can arrive from the source of information node to the rest of the nodes of the network. Note that we do not need to pre-compute μ_i, we can just run in parallel the max-path and the sum-path algorithms and stop whenever we get two consecutive coincident outputs from max-path. If we do this, Remark 2.4 guarantees that we have gone through precisely $\mu_i + 1$ iterations. Therefore, in this way, we get both a lower and an upper bound for the dispersed information.

Remark 3.2. Note also that the lower bound from the max-path algorithm is sharp (in the sense that it coincides with the dispersed information in the worst case scenario) while the upper bound from sum-path is not. This is because, for some networks, the upper bound which sum-path provides could be too rough; one of the reasons is that redundant information could be added again and again at a node when running sum-path, due to small cycles in the network. Thus, it remains an open problem, which we leave for future work, to find an efficient algorithm for computing a sharp upper bound.

3.1. **Handling several sources of information.** Until this point we have only dealt with one single source of information node. The case when there are several sources of information is more complicated as there are many possible scenarios. One possibility could be that the information at each source is just replicated from a common origin, therefore each source node holds the same amount of information M. Another different situation arises when information at two different sources is completely disjoint; note that, in this case, each source node v_i could hold a different amount of information M_i. And more complicated scenarios could appear when two sources share only part of the information stored at them. Our setting is not designed to distinguish between all this variety, as we do not take into account *which* information is lost when travelling through an edge, just *how much* of it is lost. However the algorithms we presented may be used to provide a lower and an upper bound for the transmitted information also for several sources.

To get a lower bound just run the max-path algorithm separately for each source and then take the maximum at each node among all the different outputs. It is interesting to remark that the result would be the same if a variation of the max-path algorithm is run just once with an initial vector with several non-zero coordinates, each of them representing the amount of information at the corresponding source of information node.

In order to get an upper bound, it is possible to run the sum-path algorithm separately for each source node and then sum the resulting output vectors. Note that this method could provide very rough bounds in the case that the same information is replicated along several initial nodes.

4. **An example.** In this section we will provide an example of execution of the max-path and the sum-path algorithms over the graph with 13 nodes depicted in Figure 3, which has information loss matrix

$$B = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0\\ \frac{1}{2}&0&0&0&0&0&0&0&0&0&0&0&0\\ \frac{1}{2}&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&\frac{1}{4}&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&\frac{1}{3}&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&\frac{1}{4}&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&\frac{1}{3}&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&\frac{1}{4}&0&0&0&0&0&0&0 \end{pmatrix}.$$

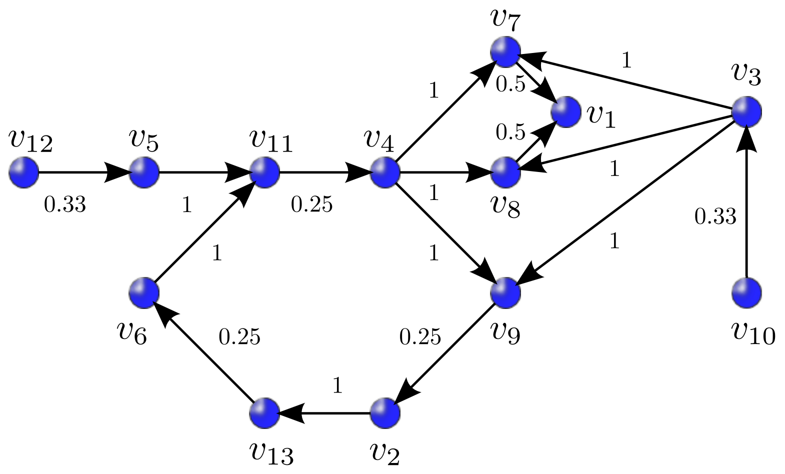

FIGURE 3. An example with 13 nodes. Source node is v_{12}. Numbers next to the links represent the proportion of information which is transmitted through that link.

We have chosen v_{12} as the source information node with an initial information amount $M = 10$. Note that the longest directed path starting at v_{12} and with no repeated vertices is the unique directed path from v_{12} to v_6, characterized by the node sequence $(v_{12}, v_5, v_{11}, v_4, v_9, v_2, v_{13}, v_6)$. As the length of this path is 7, the value μ_{12} from Remark 2.4 also equals 7, therefore the max-path algorithm should stabilize after 8 iterations. Our test run corroborated this fact.

Let us present the results of the test. We will follow the notation from Algorithm 1 and denote by x_j the output of max-path after the j-th iteration, which represents the vector of accumulated information at each node until step j; while y_j will denote the auxiliary vector used in the algorithm, which can be thought as representing the location of the information under j steps assuming that the information is travelling through the network. We will also denote by z_j the output of sum-path after the j-th iteration. The results we obtain after 8 iterations are summarized in Tables 1, 2 and 3:

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0	0	0	0	0	0.41	0.41	0.41	0.41
0	0	0	0	0	0.2	0.2	0.2	0.2
0	0	0	0	0	0	0	0	0
0	0	0	0.83	0.83	0.83	0.83	0.83	0.83
0	3.33	3.33	3.33	3.33	3.33	3.33	3.33	3.33
0	0	0	0	0	0	0	0.05	0.05
0	0	0	0	0.83	0.83	0.83	0.83	0.83
0	0	0	0	0.83	0.83	0.83	0.83	0.83
0	0	0	0	0.83	0.83	0.83	0.83	0.83
0	0	0	0	0	0	0	0	0
0	0	3.33	3.33	3.33	3.33	3.33	3.33	3.33
10	10	10	10	10	10	10	10	10
0	0	0	0	0	0	0.2	0.2	0.2

TABLE 1. Max-path output vectors

z_0	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8
0	0	0	0	0	0.83	0.83	0.83	0.83
0	0	0	0	0	0.2	0.2	0.2	0.2
0	0	0	0	0	0	0	0	0
0	0	0	0.83	0.83	0.83	0.83	0.83	0.83
0	3.33	3.33	3.33	3.33	3.33	3.33	3.33	3.33
0	0	0	0	0	0	0	0.05	0.05
0	0	0	0	0.83	0.83	0.83	0.83	0.83
0	0	0	0	0.83	0.83	0.83	0.83	0.83
0	0	0	0	0.83	0.83	0.83	0.83	0.83
0	0	0	0	0	0	0	0	0
0	0	3.33	3.33	3.33	3.33	3.33	3.33	3.38
10	10	10	10	10	10	10	10	10
0	0	0	0	0	0	0.2	0.2	0.2

TABLE 2. Sum-path output vectors

Note that, for this example, there are only two nodes, namely v_1 and v_{11} where max-path and sum-path differ. The reason behind the difference at v_1 is that 0.83

y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
0	0	0	0	0	0.41	0	0	0
0	0	0	0	0	0.2	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0.83	0	0	0	0	0
0	3.33	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.05	0
0	0	0	0	0.83	0	0	0	0
0	0	0	0	0.83	0	0	0	0
0	0	0	0	0.83	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	3.33	0	0	0	0	0	0.05
10	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.2	0	0

Table 3. Auxiliary vectors

units of information are transmitted from v_4 to v_7 and v_8 in step 4 and two halves of that information are transmitted from these two nodes to v_1 in step 5. In our setting is not possible to determine if these two halves of the information coming from v_4 are redundant or not, therefore our lower and upper bounds differ. The difference at v_{11} comes from some extra information travelling along the length six loop and returning again to v_{11}. Note also that we do not get any information at v_3 or v_{10} as there are no directed paths from v_{12} to these nodes in the graph.

5. **An application to intentional risk analysis.** Two different types of risks must be faced in *Information Technology Risk Management*: *accidental* risk and *intentional* risk. The first ones are those related to events which happen by chance, without anybody actively trying to provoke them. An example would be natural disasters. In order to manage this kind of risks, the probability of these events is estimated and some effort is made to prevent their causes. On the other hand, intentional risks are those with some active agents behind them, who are trying to achieve some kind of benefits for themselves. Game Theory has proven to be an useful tool to study, analyze and understand this kind of risks. It is remarkable that the main difference between accidental and intentional risk is that there is someone interested in producing the effect, which forces us to manage risk the other way around: we have to make the effect undesirable so that the cause will not come into being. Intentional risk management performs the opposite as accidental risk management. In accidental risk there is a cause that provokes an effect; in intentional risk, someone wishes the effect and this provokes the cause. So we have to analyze effects, and to prevent the causes we must make the effects unattractive. In any case, it is important to have in mind that the goal is not to avoid incidents but to make these incidents less profitable for the attackers.

At the present moment, the authors of this paper are working in a model for measuring the risk of suffering an intentional attack in a digital information system. In this model, a complex network is used to represent the system, where the nodes are the different components while the edges represent links between them. The *attacker* surfs on the complex network in order to get the valuable information contained in the system, but each *jump* from one node to another has its own cost depending on the characteristics of the target node and the corresponding link. Following the paradigm given by Game Theory, the focus is put on the motivating elements for the attacker. They are called *anonymity* (how easily the identity of the attacker is determined), *accessibility* (how easily the attack is carried out) and *value* (how profitable the attack is). For a more detailed discussion about these topics see, for example, [6].

In this model, an initial amount of *value* is supposed to be located at certain nodes of the network, called *vaults*. One of the examples of what *value* could be is just information, although there are other possibilities. It is also assumed that every link in the network has an associated "resistance" capability, which is quantified as a positive real number less or equal than 1. Depending on the scenario, this "resistance" can be thought as a measure of the difficulty for an attacker to get access from one node to another or as a representation of how much information located at the end node of the edge the attacker is able to access from the starting node.

This setting is a perfect fit for the max-path and sum-path algorithms we have presented, providing an useful tool to estimate how much *value* is dispersed from the *vaults* to the rest of the nodes of the network and, therefore, determine which are the most desirable nodes for the attacker.

Remark 5.1. It is important to note that, in this setting, the directions of the edges represent the path the attacker is following to reach the *value*, therefore the dispersion is made from the end node of an edge to the starting node, opposite from the convention we have used in the rest of the paper. Therefore, in step 2.2 of max-path (Algorithm 1) the matrix B should be used instead of B^T and the same change should be applied to sum-path.

6. Conclusions and future work.

In this work we have developed and proposed a model for information diffusion in a complex communication network. In our framework some amount of initial information is located at one or several information source nodes and then it travels to rest of the nodes through the links of the network. The model works under two main simplifying assumptions: first, when travelling through an edge, some information can be lost but the remaining information remains unchanged, that is, it is not modified in any way; second, we only pay attention to how much information is lost at each edge, not which information.

We have presented two efficient algorithms to compute lower an upper bounds for the amount of information received at each node of the network. The first one, which we have called max-path, provides a sharp lower bound, in the sense that gives the minimum amount of information that will arrive at each node in the more pessimistic scenario, that is, when every extra information arriving to the node is redundant with the one arrived by shorter paths. Although in the worst case (linear graph) the algorithm will take so many steps as the number of nodes in the network, this will usually be much shorter, as it equals the length of the longest directed path in the network with origin in one of the source nodes and no repeated vertices. This number does not need to be computed in order to run the algorithm, as we came up with a single stop condition, namely the output of two consecutive iterations is the same. This stop condition is optimal in the sense that stopping the algorithm at a previous step would not give the desired output.

The second algorithm, sum-path, provides an upper bound for the information disseminated at each node. As we have previously discussed this bound is not sharp and it remains an interesting open problem to find an efficient algorithm which achieves this objective. However it is important to point out that, for practical applications, there will be nodes in the network where both bounds will coincide, therefore running both algorithms in parallel with the stop condition from max-path, the exact amount of information diffused to those nodes will be computed. We have implemented both algorithms and proven them with small graphs. An example of execution is provided in Section 4. Finally we have shown how our framework and algorithms are a useful tool that can be used for intentional risk analysis.

As future work, it could be desirable to estimate the complexity of algorithms in terms of the number of links of the network, since many real systems can be modelled by using networks with sparse adjacency matrices (i.e. with a low number of links). In addition to this, a future and more detailed analysis of intentional risk will allow to study the dependence of diffusion of information process with the network topology . Recently, in [5] the authors show the importance of the position of a node in the network, so called *topocracy*, to influence the rest of the network. This can help determine which are the most desirable nodes for the attacker, or what is the node from which the attack is more effective.

Acknowledgments. The authors thank BBVA IT Risk, Fraud and Security - Research and Innovation for IT Risk, Fraud and Security, Sm4rt Security Services, Innovation for Security (I4S), Chair I4S-URJC on Information Security, Fraud Prevention and Technological Risk Management, Centro de Investigación para la Gestión Tecnológica del Riesgo (CIGTR) and University Rey Juan Carlos for their support and facilities. We also want to thank Rafael Ortega, Miguel Angel Cano, Francisco García, Juan López-Rubio, Rosa María Quintanar, Elena Alonso and Luis Solá for their useful comments and suggestions. The authors also thank the anonymous referees for their useful comments, which hugely helped us to improve the readability and references of this work. This paper was partially supported by FEDER Funds MTM2009-13848.

REFERENCES

[1] A. Barrat, M. Barthélemy and A. Vespignani, *Dynamical Processes on Complex Networks*, 1^{st} Edition, Cambridge University Press, New York, 2008.

[2] Y. Bar-Yam, *Dynamics of Complex Systems*, 1^{st} Edition, Addison-Wesley, Boston, 1997.

[3] S. Boccaletti, G. Bianconi , R. Criado, C. I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang and M. Zanin, The structure and dynamics of multilayer networks, *Physics Reports*, **544** (2014), 1–122.

[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez and D.-U. Hwang, Complex networks: Structure and dynamics, *Physics Reports*, **424** (2006), 175–308.

[5] J. Borondo, F. Borondo, C. Rodriguez-Sickert and C. A. Hidalgo, To each according to its degree: The meritocracy and topocracy of embedded markets, *Scientific Reports*, **4** (2014), 1–7.

[6] V. Chapela, *Tips for Managing Intentional Risk,* ISACA, 2011. Available from: `http://www.isaca.org/About-ISACA/-ISACA-Newsletter/`.

[7] L. R. Ford and D. R. Fulkerson, Maximal flow through a network, *Canadian Journal of Mathematics*, **8** (1956), 399–404.

[8] Y. Lin, J. C. S. Lui, K. Jung and S. Lim, Modelling multi-state diffusion process in complex networks: Theory and applications, *2013 International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 2013, 506–513.

[9] D. López-Pintado, Diffusion in complex social networks, *Games and Economic Behavior*, **62** (2008), 573–590.

[10] M. E. J. Newman, The structure and function of complex networks, *SIAM Review*, **45** (2003), 167–256.

[11] M. Safar, K. Mahdi and S. Torabi, Network robustness and irreversibility of information diffusion in Complex networks, *Journal of Computational Science*, **2** (2011), 198–206.

[12] S. H. Strogatz, Exploring complex networks, *Nature*, **410** (2001), 268–276.

Received July 2014; revised December 2014.

E-mail address: santiago.moral@bbva.com
E-mail address: vchapela@gmail.com
E-mail address: regino.criado@urjc.es
E-mail address: angel.perez@urjc.es
E-mail address: miguel.romance@urjc.es