A novel approach for fast and effective influence spread maximization in social networks

Paolo Scarabaggio, *Student Member, IEEE*, Raffaele Carli, *Member, IEEE*,
and Mariagrazia Dotoli, *Senior Member, IEEE*

Abstract—The main characteristic of social networks is their ability to quickly spread information between a large group of people. This phenomenon is generated by the social influence that individuals induce on each other. The widespread use of online social networks (e.g., Facebook) increases researchers' interest in how influence propagates through these networks. One of the most important research issues in this field is the so-called influence maximization problem, which essentially consists in selecting the most influential users (i.e., those who are able to maximize the spread of influence through the social network). Due to its practical importance in various applications (e.g., viral marketing, target advertisement, personalized recommendation), such a problem has been studied in several variants. Different solution methodologies have been proposed. Nevertheless, the current open challenge in the resolution of the influence maximization problem still concerns achieving a good trade-off between accuracy and computational time. In this context, based on the well-known independent cascade and the linear threshold models of social networks, we propose a novel low-complexity and highly accurate algorithm for selecting an initial group of nodes to maximize the spread of influence in large-scale networks. In particular, the key idea consists in iteratively removing the overlap of influence spread induced by different seed nodes. Application to several numerical experiments based on real datasets proves that the proposed algorithm effectively finds practical near-optimal solutions of the addressed influence maximization problem in a computationally efficient fashion. Finally, comparison with the best performing state of the art algorithms demonstrates that in large scale scenarios, the proposed approach shows higher performance in terms of influence spread and running time.

Index Terms—Social networks, Independent cascade model, Linear threshold model, Activation probability computation, Influence maximization.

I. INTRODUCTION

A social network is a structure composed of a group of interconnected independent agents. These networks are essential for the spread of information, opinion, and innovation. Several studies have been made in social science to correctly evaluate the spread of influence, especially related

to the “word-of-mouth” process [1], [2]. The core idea of this process is the influence between people in making actions, i.e., a person may hear about a specific innovation and decide to implement it too. This diffusion process is essential for viral marketing, personalized recommendation, and target advertisement. Therefore, attention is increasing on how influence propagates in online social networks [3], [4]. For instance, in viral marketing, the best way to spread information about a product is to select a group of users that have a strong influence on the network and convince them to promote it. Due to their high influence, many of their neighbors will share information, causing a cascade process. Naturally, this process is fruitful if, with a low number of initial agents, it is possible to significantly influence the network.

The study of the diffusion process in a network can be done employing different models; in the related literature, two models are particularly examined for their effectiveness: the linear threshold (LT) and the independent cascade (IC) model [5]. Both models comprehend active and inactive nodes: the former have been already influenced by the information spreading in the network and consequently attempt to spread it to their neighbors in a cascade process. In the LT model, the inactive nodes receive information when a large number of their neighbors are active. More formally, each inactive node has a threshold in the interval $[0, 1]$, which expresses the fraction of neighbors that should be active in order to receive information [6]. Conversely, in the IC model each node tries to directly influence its inactive neighbors. The success of this activation process is ruled by a probability associated with the edge; however, each node has only one chance to influence its neighbors [7].

Several works analyze the probability related to the diffusion of information in social networks [8]–[10]. In particular, considerable attention is focused on determining the set of active nodes at the end of the diffusion process. Both the IC and the LT models allow estimating the influence spread by summing up all the nodes' activation probabilities; these represent the probability that a single node gets influenced. Nevertheless, computing the precise influence spread under the IC and LT models is #P-hard [11], [12]. Hence, Monte Carlo simulation is applied in many works, even if it is time-consuming [11], [13], [14]. To cope with this issue, the authors in [15] propose the *SteadyStateSpread* approximated algorithm to compute an estimation of the influence spread; however, the value calculated by the algorithm may be considerably different from the exact solutions of the problem, depending

Manuscript received October 23, 2020. (Corresponding author: Paolo Scarabaggio.)

P. Scarabaggio, R. Carli, and M. Dotoli are with the Department of Electrical and Information Engineering of the Polytechnic of Bari, Italy. (e-mail: paolo.scarabaggio, raffaele.carli, mariagrazia.dotoli@poliba.it).

The work of M. Dotoli is partially supported by the Italian University and Research Ministry under project RAFAEL (National Research Program, contract n. ARS01_00305).

on the network structure.

An additional problem in these networks is selecting a group of initial nodes that spread information at most. This second problem was firstly addressed in [16], [17]. In [18], the authors define a greedy algorithm able to solve the influence maximization problem. The influence maximization problem is computationally hard. In fact, given the request for a significant computational effort, many works aim at decreasing the running time [18]. Most of these works use heuristic approaches, for instance by assuming that all nodes have relatively low influence or that the input graph is clustered [11], [19], [20]. Other works define the influence spread as a submodular process [14], [21]. Furthermore, some works attempt to solve the influence maximization problem by considering the concept of influence overlap. The influential nodes in social networks usually are concentrated [22], in such a way that their resulting influences overlap, generating a redundancy [23]. Recent studies have paid attention to reducing influence overlap by keeping nodes disconnected [24]. The problem of influence overlap is optimally solved by employing the greedy algorithm proposed in [18], which ensures an almost optimal convergence; however, this method is highly time-consuming and is not feasible for large scale networks.

Summing up, the previous literature review shows that the current open challenge in the study of the influence diffusion process in social networks still concerns achieving a good trade-off between accuracy and computational time.

To cope with this gap, first of all we propose a novel low-complexity and highly accurate algorithm for influence spread computation and maximization in large-scale networks. In particular, the proposed approach is based on the reduction of influence overlapping and leverages on the joint probability of activations related to different nodes occurring together. At each iteration, the algorithm chooses a node to include in the seed set and then attempts to estimate and remove the influence overlap with the other candidate nodes. The approach is valid both for the IC and LT diffusion model.

More in detail, the specific contributions of this work are summarized in the following aspects.

- For the sake of exactly computing the influence spread in a social network based on the LT diffusion model, we extend to the LC case the Path Method proposed by [25] for the IC diffusion model only.
- Leveraging on the *SteadyStateSpread* approach proposed by [15] to iteratively calculate the influence spread in IC networks, we present its counterpart for LT models, proving its convergence to a fixed point.
- We propose a novel approach for influence maximization based on the reduction of influence overlapping and utilization of the joint probability of activations to efficiently estimate the marginal gain in the influence spread computation without recalculating it at each iteration.
- We show how the level of spread probability and the topology of network graphs influence the accuracy of results in terms both of influence spread computation and influence maximization.
- We examine different combinations of algorithms for the activation probability computation with the proposed

algorithm to select the seed set and solve the influence maximization problem.

- We show through several numerical experiments based on real datasets that the proposed algorithm effectively finds practical near-optimal solutions of the addressed influence maximization problem in a computationally efficient fashion. Finally, the comparison with other state-of-the-art mechanisms (including the *Greedy* algorithm) demonstrates that the proposed approach shows higher performance in terms of influence spread and running time in large scale scenarios.

The rest of this work is structured as follows. In Section II we present the social network model and some preliminary definitions. In Section III we present two influence diffusion models and show the procedures to compute each node's activation probability. Section IV comprehends the formulation of the influence maximization problem and the definition of the best performing algorithms that are available in the related literature to solve it; moreover, we propose a novel algorithm to solve such a problem. In Section V we show the numerical results of the conducted experiments. Lastly, in Section VI we provide the outcomes of this work.

II. PRELIMINARIES

Graphs are widely used to represent networked systems [26], [27]. Thus, let us define a social network by a direct graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes with cardinality $N = |\mathcal{V}|$ representing people, and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the set of edges describing the social connections between pairs of people. In the sequel, we refer to nodes and edges, putting aside the words people and social connections.

When node j is influenced by node i , an edge denoted by $(i, j) \in \mathcal{E}$ exists. Finally, let us define by $N_j^{\text{in}} = \{i \in \mathcal{V} | (i, j) \in \mathcal{E}\}$ the in-neighbors of node j and $N_j^{\text{out}} = \{i \in \mathcal{V} | (j, i) \in \mathcal{E}\}$ the out-neighbors of node j .

Let us now recall from [28] some definitions used in the rest of this paper.

Definition 1: Given a social network model, with direct graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we describe the *diffusion model* (DM) as the stochastic process that captures the spread on the information in the network starting from a seed set $\phi_0 \subseteq \mathcal{V}$ that contains the initially activated nodes.

Definition 2: For a given diffusion model $\mathcal{G}_{\text{DM}} = (\mathcal{V}, \mathcal{E}, \text{DM})$ and a seed set $\phi_0 \subseteq \mathcal{V}$, the probability that a node $j \in \mathcal{V}$ is activated during the diffusion process is defined as its *activation probability* $\pi_j(\phi_0)$.

Definition 3: For a given diffusion model $\mathcal{G}_{\text{DM}} = (\mathcal{V}, \mathcal{E}, \text{DM})$ and a seed set $\phi_0 \subseteq \mathcal{V}$, the *influence spread* across the network $\sigma(\phi_0)$, is defined as follows:

$$\sigma(\phi_0) = \sum_{j \in \mathcal{V}} \pi_j(\phi_0) \quad (1)$$

Definition 4: Assuming that $\sigma(\cdot)$ is a non-negative set function, we have that for every possible seed set $\phi_A \subseteq \mathcal{V}$ the quantity $\Delta_n(\phi_A) = \sigma(\phi_A \cup \{n\}) - \sigma(\phi_A)$ is called the *marginal gain* of node $n \in \mathcal{V} \setminus \phi_A$ with respect to set ϕ_A . Similarly, we define the marginal gain of the set $\phi_B \subseteq \mathcal{V} \setminus \phi_A$ as $\Delta_{\phi_B}(\phi_A) = \sigma(\phi_A \cup \phi_B) - \sigma(\phi_A)$.

Theorem 1 ([29]): Assuming that $\sigma(\cdot)$ is a non-negative set function, we have that $\sigma(\cdot)$ is monotone if $\sigma(\phi_A) \geq \sigma(\phi_B)$ for every $\phi_B \subseteq \phi_A \subseteq \mathcal{V}$.

Theorem 2 ([29]): Assuming that $\sigma(\cdot)$ is a non-negative set function, we have that $\sigma(\cdot)$ is submodular if $\Delta_n(\phi_A) \leq \Delta_n(\phi_B)$ for every $\phi_B \subseteq \phi_A \subseteq \mathcal{V}$.

III. INFLUENCE DIFFUSION MODELS

Several models are available in the literature to represent the spread of information in a network. These models rely on different mechanisms to capture the switching mechanism from inactive to active node status. This section focuses on the two most common diffusion models: the Independent Cascade (IC) model and the Linear Threshold (LT) model.

A. Independent Cascade Model

The spread of influence through social networks is extremely well studied, and several models are available in the literature. The IC model is one of the most studied probabilistic information diffusion models and is based on the assumption that information flows over the network through cascades. This means that a node can be triggered from specific information only if one of its in-neighbors was triggered before. In the IC model, nodes can have two states: active (i.e., already influenced by the information) or inactive (i.e., not influenced). The process runs over discrete steps. Initially, i.e., at time step $k = 0$, most nodes are inactive; only a few nodes are active and are known as seed nodes. Let us denote as $\phi_0 \subseteq \mathcal{V}$ the seed set containing all the initially active nodes. If node i becomes active at step k , it has one and only one chance to activate its inactive neighbors $j \in N_i^{out}$ with a propagation probability $p_{i,j}$. The propagation probabilities $p_{i,j} \forall (i,j)$ measure the influence strength of the edges; they are collected in matrix $P : (\mathcal{V} \times \mathcal{V}) \rightarrow [0, 1]$. It is evident that $p_{i,j} = 0$ if there is no edge (i,j) in $\mathcal{E}$.

If node i successfully activates one of its out-neighbors $j \in N_i^{out}$ at time k , then j will become an active node at the next time step $k + 1$. However, note that a node can change its state from inactive to active but not from active to inactive. This cascading process continues until no more activation occurs through the network [30]. More formally, we can denote the considered IC model by $\mathcal{G}_{IC} = (\mathcal{V}, \mathcal{E}, P)$.

B. The Linear Threshold Model

Together with the IC model, the LT model is one of the most studied diffusion models. It is used in several variants, however, the most common is the so-called submodular LT model. In such a model the basic idea is that a user can switch its status from inactive to active if a sufficient number of its incoming neighbors are active. Similarly to the IC model, nodes can have two states: active (i.e., already influenced by the information) or inactive (i.e., not influenced). The process runs over discrete steps. Just like in the IC model, initially, i.e., at time $k = 0$, all nodes are inactive, except for a few nodes which are contained in the seed set $\phi_0 \subseteq \mathcal{V}$. A weight $b_{i,j}$ - that measures the influence strength of the edges - is

associated to each edge (i,j) ; all the weights are collected in matrix $B : (\mathcal{V} \times \mathcal{V}) \rightarrow [0, 1]$. Similarly to the IC model, it is evident that $b_{i,j} = 0$ if there is no edge (i,j) in $\mathcal{E}$. However, the submodular LT model requires that $\sum_{N_j^{in}} b_{i,j} \leq 1 \forall j \in \mathcal{V}$, i.e., for each node the sum of weights of all its in-neighbors must be lower than one.

Considering an instance of the diffusion process, the LT model first samples for each node n a threshold value $\theta_n \in [0, 1]$ randomly. Then, at step k , all nodes that were active in step $k - 1$ remain active, and any node n that was inactive in step $k - 1$ switches to active if the total weight of its neighbors that became active at step $k - 1$ is at least $\theta_n(k)$. More in detail, the activation process will be successful if the sum of the incoming active neighbor's weights becomes either greater than or equal to the node's threshold, i.e.,

$$\sum_{\forall i \in N_j^{in}(k-1)} b_{i,j} \geq \theta_j(k) \quad (2)$$

where $N_j^{in}(k-1)$ denotes the set of in-neighbors of j activated at the step $k - 1$. This method will be continued until no more activation is possible. More formally, we can denote the considered LT model by $\mathcal{G}_{LT} = (\mathcal{V}, \mathcal{E}, B)$.

C. Exact Activation Probability Computation

In [25] the authors propose the *Path Method*, an algorithm to exactly compute the value of $\pi_j(\phi_0)$ for IC networks. This algorithm takes into account all the possible evolutions of the information spread in a social network. In the first part, an evolution graph for the model is created; this is made of cells shown in Fig. 1(a). Each cell is divided into three parts: the set of past active nodes A_k^p containing all the nodes activated before the current step k , the set of active nodes A_k^c at time k and the cell probability P_k (i.e., the probability of the evolution contained in A_k^p and A_k^c). The cells that have an empty current active set are the terminal cells. The exact activation probability of node j can be computed by summing up all the cell probabilities of terminal cells whose set of past active nodes contains node j . For instance, we show such a procedure in reference to the simple IC model in Fig. 1(b), where the corresponding directed graph is composed of three nodes. In particular, Fig. 2 shows the resolution steps of the *Path Method* for the aforementioned example, when the seed set includes only node 1, i.e., $\phi_0 = \{1\}$. In this example, we show the graph's evolution through discrete time steps, and we indicate in red the terminal cells. For instance, if we consider node 3, its activation probability $\pi_3(\{1\})$ is the sum of all the probabilities P_k of the terminal cells whose sets A_k^c and A_k^p contain the node 3; namely $\pi_3(\{1\}) = 0.06 + 0.07 + 0.216 + 0.024 = 0.37$.

The path method was explicitly studied for the IC diffusion model; however, it can be easily modified to take into account the LT model, as shown in Fig. 3 for the aforementioned example.

Due to the high complexity of the *Path Method*, its computational time is $O(6^N)$ [25]. The exponential complexity of this approach makes it not feasible for large-scale networks.

Fig. 1. (a) Cell of the evolution graph. (b) Example of a social network model with 3 nodes.

Fig. 2. Evolution graph related to the network shown in Fig. 1(b) with the IC diffusion model.

Fig. 3. Evolution graph related to the network shown in Fig. 1(b) with the LT diffusion model.

As aforementioned, the *Path Method* is able to calculate the exact activation probability of each node; however, it may be infeasible even for small networks. An alternative approach to estimate the exact activation probability is the use of the well-known *Monte Carlo* technique. In fact, the IC and the LT models are stochastic diffusion processes that each time lead to different outcomes. However, on average, the final results will represent the exact activation probabilities. More in detail, for each run r of the stochastic diffusion model, let us define by $\pi_j^r(\phi_0)$ the corresponding result; then, by performing R runs, we can estimate the activation probabilities by:

$$\bar{\pi}_j(\phi_0) = \mathbb{E}[\pi_j(\phi_0)] = \frac{\sum_{r=1}^R \pi_j^r(\phi_0)}{R}. \quad (3)$$

This approach's reliability increases with the number of instances; however, the required computational time makes it practically infeasible for large scale networks.

D. Approximate Activation Probability Computation

As mentioned, the activation probability computation is a hard problem to be solved because it cannot be easily expressed in a closed form. In [15] the authors propose the *SteadyStateSpread* (SSS) algorithm, which is able to estimate

the activation probability of each node. In the given network, node j may receive the influence by one of its in-neighbors, each with a specific activation probability. Therefore, each node does not assimilate information when none of its in-neighbors transmits it. Assuming that the activations of the in-neighbors are independent events and they do not depend on the activation of node j , we calculate the probability that none of the in-neighbors of node j transmits the information by:

$$\prod_{i \in N_j^{in}} (1 - \pi_i(\phi_0) p_{i,j}). \quad (4)$$

Thus, we have:

$$1 - \pi_j(\phi_0) = \prod_{i \in N_j^{in}} (1 - \pi_i(\phi_0) p_{i,j}). \quad (5)$$

As proposed in [15], the computation of the activation probability of all the nodes in a network can be done iteratively base on (5) by setting:

$$\pi_j^+(\phi_0) = \begin{cases} 1 & \text{if } j \in \phi_0 \\ 1 - \prod_{i \in N_j^{in}} (1 - \pi_i(\phi_0) p_{i,j}) & \text{if } j \notin \phi_0 \end{cases} \quad (6)$$

where we mark with the plus symbol (+) the updated value of $\pi_j(\cdot)$ at each iteration.

However, the SSS approach is able to compute the activation probability of all the nodes in a network only when the diffusion model is the IC. In fact, this approach does not apply to the LT models, where different assumptions hold. To cope with this issue, let us propose a novel approach - inspired by the classic SSS - named *Linear Threshold SteadyStateSpread* (LT-SSS). Similarly to the classic SSS, also this approach iteratively calculates all the nodes' activation probability in a network:

$$\pi_j^+(\phi_0) = \begin{cases} 1 & \text{if } j \in \phi_0 \\ 1 - \left(1 - \sum_{i \in N_j^{in}} (\pi_i(\phi_0) b_{i,j}) \right) & \text{if } j \notin \phi_0. \end{cases} \quad (7)$$

The approaches described above does not lead to the computation of the exact activation probability, due to the assumption made on the independence of in-neighbor activation events [13]. The time complexity of this approach is $O(N^2)$ [15].

Theorem 3: If a_n is a monotone sequence of real numbers, then a_n is convergent if and only if a_n is bounded.

Proposition 1: The sequence generated by the LT-SSS algorithm, i.e., the system (7) converges to a unique fixed-point.

Proof: Let us assume that the activation probabilities $\pi_i(\phi_0) \forall i \in \mathcal{V}$ are always positive and bounded in the interval $[0,1]$. As we defined the weights as $\sum_{i \in N_j^{in}} b_{i,j} \leq 1$ the activation probability for every node at any subsequent step $\pi_j^+(\phi_0)$ is also bounded in $[0,1]$ and the following relation holds: $\pi_j^+(\phi_0) \geq \pi_j(\phi_0)$. Therefore, it is just necessary to show that the value of $\pi_i(\phi_0) \forall i \in \mathcal{V}$ at the first iteration is bounded in $[0,1]$, as it is in the seed set definitions:

$$\pi_j^0(\phi_0) = \begin{cases} 1 & \text{if } j \in \phi_0 \\ 0 & \text{if } j \notin \phi_0. \end{cases} \quad (8)$$

IV. FORMULATION AND RESOLUTION OF THE INFLUENCE MAXIMIZATION PROBLEM

Let us now formulate the influence maximization problem, given a diffusion model, the definition of information spread, and an algorithm to compute an estimation for the activation probabilities. The problem is to find S initial nodes that maximize the influence spread through the network. More formally, we can define the maximization problem as:

$$\phi_0 = \operatorname{argmax}_{\phi \subseteq \mathcal{V}} \{\sigma(\phi) \mid |\phi| = S\} \quad (9)$$

where S is the chosen cardinality for the seed set. Several algorithms are available in the related literature to approximately solve this problem [31]. In this section we first recall two well-known approaches (namely the *SelectTopS* and the *Greedy* algorithm); subsequently, we present two novel algorithms to effectively find near-optimal solutions to the influence maximization problem in a computationally efficient fashion.

A. SelectTopS Algorithm

The easiest approach to solve the maximization problem is to consider each node i as an initial seed set $\phi_0 = \{i\}$. Then it is possible to compute the activation probability $\pi_j(\{i\})$ of each node $j \in \mathcal{V}$, and select the S nodes that gave the higher influence spread $\sigma(\{i\}) = \sum_{j \in \mathcal{V}} \pi_j(\{i\})$. The formal definition of the *SelectTopS* algorithm is reported in Algorithm 1 [28].

Let us name T the time complexity needed to calculate the activation probability across a given network of N nodes. Since the calculation of the influence spread for every single node being a singleton seed set is needed, it is straightforward noting that the time complexity of the *SelectTopS* algorithm is $O(NT)$.

B. Greedy Algorithm

Let us now recall the *Greedy* algorithm presented in [18].

The definition of *Greedy* algorithm is reported in Algorithm 2 and is summarized in the sequel. The algorithm starts with the empty seed set ϕ_0 , and then iteratively selects a node which is not currently in ϕ_0 , whose inclusion implies the highest marginal increment in $\sigma(\cdot)$. The process of recalculating the activation probability for the entire network for each node in the final seed set causes the algorithm's time inefficiency. Recalling that T is the time complexity required to compute the activation probability across the given network with N nodes, the time complexity for the *Greedy* algorithm is $O(SNT)$ [18].

This algorithm guarantees, under the conditions described in the following theorem, that the obtained influence spread approximates the optimal value by a factor $(1 - 1/e)$, where e denotes the base of the natural logarithm; however, the algorithm presents a significant computational time.

Theorem 4 ([18]): For a non-negative, monotone, and sub-modular influence function $\sigma(\cdot)$, let ϕ_0 be a set of size S obtained by the greedy algorithm. Then ϕ_0 satisfies $\sigma(\phi_0) \geq (1 - 1/e)\sigma(\phi_0^*)$ where ϕ_0^* is an optimum solution.

Algorithm 1 *SelectTopS* algorithm [28]

Input: $\mathcal{G}_{IC}$, $S \in \mathbb{N}^+$

- 1: **for** $i \in \mathcal{V}$ **do**
- 2: **for** $j \in \mathcal{V}$ **do**
- 3: Compute $\pi_j(\{i\})$
- 4: **end for**
- 5: Compute $\sigma(\{i\}) = \sum_{j \in \mathcal{V}} \pi_j(\{i\})$
- 6: **end for**
- 7: Select S nodes n with the highest value of $\sigma(\{n\})$

Output: ϕ_0

Algorithm 2 *Greedy* algorithm [18]

Input: $\mathcal{G}_{IC}$, $S \in \mathbb{N}^+$

- 1: Initialize $\phi_0 \leftarrow \emptyset$
- 2: **for** $s = 1$ to S **do**
- 3: **for** $i \in \mathcal{V} \setminus \phi_0$ **do**
- 4: **for** $j \in \mathcal{V}$ **do**
- 5: Compute $\pi_j(\phi_0 \cup \{i\})$
- 6: **end for**
- 7: Compute $\sigma(\phi_0 \cup \{i\}) = \sum_{j \in \mathcal{V}} \pi_j(\phi_0 \cup \{i\})$
- 8: **end for**
- 9: Select $n = \operatorname{argmax}_{i \in \mathcal{V} \setminus \phi_0} \{\sigma(\phi_0 \cup \{i\}) - \sigma(\phi_0)\}$
- 10: $\phi_0 \leftarrow \phi_0 \cup \{n\}$
- 11: **end for**

Output: ϕ_0

Theorem 5 ([32], [33]): $\sigma(\cdot)$ is a monotone (non-decreasing) submodular set function if and only if:

$$\sigma(S_1) \leq \sigma(S_2) + \sum_{i \in S_1 \setminus S_2} \rho_i(S_2). \quad (10)$$

Corollary 1 ([18]): For an arbitrary instance of the IC model or the submodular LT model, the resulting influence function $\sigma(\cdot)$ is monotone and submodular.

C. The Novel Proposed Algorithm for Influence Maximization

As described above, the *Greedy* algorithm requires a high computational time, which increases linearly over the size of the final seed set S . In this paper, we propose a novel algorithm based on approximately maximizing the submodular function through an iterative selection of nodes for the final seed set. The proposed approach, denoted as *Reduced Influence Overlap (RIO)* algorithm, is able to solve the influence maximization problem within a shorter computational time, while providing a good approximation of the optimal solution.

First, we describe the core idea of the proposed algorithm referring to the simple graph shown in Fig. 4 under the IC model, where node c may be influenced by node a and node b . It is apparent that the activation probability of node c is $\pi_c(\{a\})$ and $\pi_c(\{b\})$ when a and b is the initial node, respectively.

Assuming that a is the only node in the seed set, i.e., $\phi_0 = \{a\}$ and $\pi_c(\phi_0) = \pi_c(\{a\})$, we want to evaluate the new activation probability when b is added to the seed set. In this

Fig. 4. Example of social network used to describe the core idea of the proposed algorithm.

simple example, such a value can be calculated as:

$$\pi_c(\phi_0 \cup \{b\}) = 1 - (1 - \tilde{\pi}_c(\phi_0))(1 - \pi_c(\{b\})). \quad (11)$$

However, we highlight that the proposed algorithm is based on the assumption that the activations of the in-neighbors of node j are independent events. Hence, equation (11) is not accurate in social networks due to the non-independence between nodes. Nevertheless, (11) gives a good basis to develop an estimate on how the activation probabilities change when a node is included in the seed set.

In the example mentioned above, by means of (11), we can exactly compute the new activation probability without having a second run of the activation spread algorithm (e.g., as in the *Path Method*, *SteadyStateSpread* or *Monte Carlo* approach) because the two activation events are independent. However, by analyzing the network in Fig. 1 with the IC model, we do not have the assumption on the independency of the activation events. The node with the highest influence spread is obviously the first, that has a $\pi_2(\{1\}) = 0.224$ and a $\pi_3(\{1\}) = 0.370$. Consequently, this is the first node to be added to the seed set. Next, we aim to evaluate the activation probability of node 3 when node 2 is included in the seed set, i.e., $\pi_3(\{1\} \cup \{2\})$, without running another time the influence spread algorithm (i.e., like in the *Greedy* algorithm). By employing (11) we can use $\pi_3(\{1\})$ and $\pi_3(\{2\})$ to estimate $\pi_3(\{1\} \cup \{2\}) \approx \tilde{\pi}_3(\{1\} \cup \{2\}) = 1 - (1 - 0.224)(1 - 0.370) = 0.511$. Let us point out that the real value obtained with the *Greedy* approach is $\pi_3(\{1\} \cup \{2\}) = 0.650$, hence about 20% higher than by the approximated approach.

The definition of the *RIO* algorithm is reported in Algorithm 3 and is summarized in the sequel. In the first phase (Algorithm 3, lines 1-5), following the *SelectTopS* algorithm, we calculate the activation probability $\pi_j(\{i\})$ of each node $j \in \mathcal{V}$ by considering each node i as a seed set. Moreover, we initialize the seed set ϕ_0 to the empty set and $\tilde{\sigma}(\phi_0)$ and $\tilde{\pi}_j(\phi_0)$ to zero (Algorithm 3, lines 6-9). In the second phase (Algorithm 3, lines 9-16), we estimate the new activation probability of all the nodes in the network by including an additional node in the seed set by means of (11) and we compute the new value of the influence spread function. Next, we select the seed node n with the highest expected gain of information spread, and we include it in the final seed set ϕ_0 . In the last phase (Algorithm 3, lines 17-18), we update the activation probability of all the node $j \in \mathcal{V}$ when the node $i \in \mathcal{V} \setminus \phi_0$ is the seed node, the influence spread function and the activation probability.

Finally, we provide the complexity analysis for the proposed algorithm. It is apparent that the complexity of the *RIO* algorithm is dominated by the first phase (where the activation

Algorithm 3 *RIO* algorithm

Input: $\mathcal{G}_{IC}$, $S \in \mathbb{N}^+$

```

1: for  $i \in \mathcal{V}$  do
2:   for  $j \in \mathcal{V}$  do
3:     Compute  $\pi_j(\{i\})$ 
4:   end for
5: end for
6: Initialize  $\phi_0 \leftarrow \emptyset$ 
7: Initialize  $\tilde{\sigma}(\phi_0) \leftarrow 0$ 
8: Initialize  $\tilde{\pi}_j(\phi_0) \leftarrow 0 \quad \forall j$ 
9: for  $s = 1$  to  $S$  do
10:  for  $i \in \mathcal{V} \setminus \phi_0$  do
11:    for  $j \in \mathcal{V}$  do
12:       $\tilde{\pi}_j(\phi_0 \cup \{i\}) \leftarrow 1 - (1 - \tilde{\pi}_j(\phi_0))(1 - \pi_j(\{i\}))$ 
13:    end for
14:    Compute  $\tilde{\sigma}(\phi_0 \cup \{i\}) = \sum_{j \in \mathcal{V}} \tilde{\pi}_j(\phi_0 \cup \{i\})$ 
15:  end for
16:  Select  $n = \underset{i \in \mathcal{V} \setminus \phi_0}{\operatorname{argmax}} \{ \tilde{\sigma}(\phi_0 \cup \{i\}) - \tilde{\sigma}(\phi_0) \}$ 
17:   $\phi_0 \leftarrow \phi_0 \cup \{n\}$ 
18:   $\tilde{\sigma}(\phi_0) \leftarrow \tilde{\sigma}(\phi_0 \cup \{n\})$ 
19:   $\tilde{\pi}_j(\phi_0) \leftarrow \tilde{\pi}_j(\phi_0 \cup \{n\}) \quad \forall j$ 
20: end for
Output:  $\phi_0$ 

```

probabilities are determined) only, whilst the second and third phases contain negligible operations. Therefore, recalling that T denotes the time complexity for calculating the activation probability across the given network of N nodes, the time complexity of the *RIO* algorithm is straightforwardly equal to $O(NT)$.

V. NUMERICAL EXPERIMENTS

This section on numerical experiments is divided into three parts. In the first part, we present the datasets. Then, we analyze the activation probability computation: in particular, we compare the influence spread value obtained with the SSS and the LT-SSS approaches with respect to the Monte Carlo simulation results. Lastly, we evaluate the proposed *RIO* algorithm in terms of the seed set selection and computational time.

A. Datasets Description

In our experiments, we adopt two sets of regular networks and two public real network datasets [34], [35].

The first set of networks is a series of rings composed of 8 nodes connected in a circle; each node is connected symmetrically with its neighbors. Let us define the network parameter c that indicates the number of connected neighbors, for instance, we show a ring with $c = 4$ in 2 Fig. 5(a). We assume $p_{i,j} = b_{i,j} = 0.1$ for all edges. It should be noted that, being at most $c = 6$, we have $\sum_{N_j^{in}} b_{i,j} = 0.6 \leq 1$ that still preserves the submodularity of the LT model. The second synthetic network is a bidirectional grid graph of 16 nodes. For this network we define an additional parameter $d \in [0, 1]$ to describe the level of influence spread probability in the overall

Fig. 5. Examples of the regular networks used in the simulations: ring network with 8 nodes and $c = 4$ (a); grid network with 16 nodes (b).

network. More in detail, we calculate the probability for the IC model and the weights for the LT model as $p_{i,j} = d/N_j^{\text{in}}$ and $b_{i,j} = d/N_j^{\text{in}}$, where N_j^{in} are the numbers of in-neighbors of node j .

The first real dataset is called Hamsterster (1858 nodes, 12534 edges) and is related to the homonym website created in 2003. The website worked as a social network for pets, where their owners create friendships between the animals. In this dataset, pets are the nodes, while their friendships are the edges. We adopt the IC and the LT models to simulate the information diffusion process; however, since we do not have the data to infer weights of edges, we use the *Trivalency* approach in [11] to randomly assign the probability for the IC model by selecting $p_{i,j}$ from the discrete set $\{0.01, 0.1, 0.5\}$, moreover, to ensure the submodularity in the LT model, we calculate the weights as $b_{i,j} = \frac{p_{i,j}}{\sum_{j \in \mathcal{V}} p_{i,j}} r$ where $r \in [0, 1]$ is a random value.

Second, we use the ego-Facebook real dataset (4039 nodes and 88234 edges). This dataset consists of anonymized friends lists from Facebook. The dataset includes nodes (profiles), circles, and ego networks. Data are anonymous. Moreover, feature vectors from this dataset have been provided, even though the interpretation of those features has been obscured. For instance, if the original dataset contained a feature "political=Democratic Party", the new data would contain "political=feature 1". Thus, using the anonymized data it is possible to determine whether two users have the same political affiliations. The dataset does not contain any direct information related to the strength of the social connection between two different users: however, we estimate this value by employing the anonymized features, i.e., the most features two nodes share, the strongest their connection.

B. Activation Probability Computation

In the first part of the simulations, we analyze the reliability of the approximated approach (i.e., the SSS and the LT-SSS) in computing the activation probability. More in detail, we evaluate the impact of the influence spread level and the grid topology by changing the parameters c and d of the two regular networks. In order to correctly assess the performance of the proposed approximated approach, we compare the corresponding results with those achieved by the *Monte Carlo* method over 10000 runs. Adding one node per time in the seed set, we evaluate the value of the influence spread calculated by the SSS and the *Monte Carlo* method. However, the relative errors between these two techniques are highly dependent on

Fig. 6. Relative error of the SSS in the ring network with $p_{i,j} = b_{i,j} = 0.1 \forall i, j \in \mathcal{E}$: IC model with $d = 2$ (a), IC model with $d = 4$ (b), IC model with $d = 6$ (c), LT model with $d = 2$ (d), LT model with $d = 4$ (e), and LT model with $d = 6$ (f). The shaded areas represent the error for the whole set of experiments; the solid lines represent the average error.

Fig. 7. Relative error of the SSS in the grid network: IC model with $d = 0.3$ (a), IC model with $d = 0.5$ (b), IC model with $d = 0.7$ (c), LT model with $d = 0.3$ (d), LT model with $d = 0.5$ (e), and LT model with $d = 0.7$ (f). The shaded areas represent the error for the whole set of experiments; the solid lines represent the average error.

the order used to include the nodes in the seed set. Therefore, we perform a simulation per every combination and calculate the confidence interval for the resulting relative error.

In Fig. 6 we show the results corresponding to ring network in Fig. 5(a) with $c \in \{2, 4, 6\}$ and using the IC and the LC modeling. It is apparent that the relative error of the SSS and the LT-SSS approaches with respect to the *Monte Carlo* method increase when additional connections are included in the network. In fact, by increasing the connections, the hypothesis on the independence of the activation events become weaker.

Moreover, Fig. 7 shows the results corresponding to the grid network Fig. 5(a) when $d \in \{0.3, 0.5, 0.7\}$. In this case, the increase of the level of probability spread in the network impact negatively the accuracy of the approximated approaches.

It should be noted that the LT-SSS shows a higher relative error with respect to its IC counterpart both in the ring and grid networks simulations. These experiments indeed employ networks that are considerably different from real social networks. In fact, social networks are characterized by a large number of nodes and relatively small propagation probability [13]. Moreover, the influence spread is clustered; therefore, a

Fig. 8. Influence spread computed by the *SteadyStateSpread* (solid line) and the *Monte Carlo* (dashed line) approaches: Hamsterster and IC model (a), Hamsterster and LT model (b), ego-Facebook and IC model (c), and ego-Facebook and LT model (d).

local error would not profoundly affect the influence spread function. As a proof of this, in Fig. 8 we show the influence spread for the two real networks computed by the *SteadyStateSpread* and the *Monte Carlo* approach. It is evident that the approximated approaches are highly accurate both in the IC and LT models.

C. Influence Maximization and Nodes Selection

In the second part of the simulations, we examine the influence maximization problem, assessing the performance of the *RIO* algorithm. In particular, we show the influence spread given by the nodes selected by different state-of-the-art algorithms. The experiment is divided into two parts. In the first part, we select the seed nodes by employing one of the algorithms mentioned above that may approximate the influence spread function (e.g., the *RIO* algorithm). After selecting the seed nodes, we evaluate the real influence spread by performing a Monte Carlo simulation with 10000 runs for each selected seed set by the different algorithms.

In Fig. 9 we show the influence spread for the ring network with $c \in \{2, 4, 6\}$, and with the IC and the LC models. In this case, the increased c reduces the difference between the *SelectTopS* and the *RIO*. However, the differences between the *RIO* and the *Greedy* are always small even if the hypothesis on the activation events' independency is not met.

Moreover, Fig. 10 shows the results of the grid network when $d \in \{0.3, 0.5, 0.7\}$. In this case, a higher level of probability spread in the network makes the *RIO* algorithm results poorer than the *Greedy*.

The first simulations using the artificial datasets cannot fully show the proposed approach's advantage due to the small number of nodes. Therefore, let us show the results obtained with the real datasets. Similarly, the simulations are made in two parts. In the first, a combination of algorithms for the influence spread computation and the influence maximization is used to select the most influential users. Then in the second part, the obtained seed sets are validated with a Monte Carlo simulation. Let us remark that we need the validation phase because the influence spread computed with the *SteadyState-*

Fig. 9. Influence spread computed by the tested algorithms in the ring network with $p_{i,j} = b_{i,j} = 0.15$, $\forall i, j \in \mathcal{E}$: $c = 2$ and IC model (a), $c = 4$ and IC model (b), $c = 6$ and IC model (c), $c = 2$ and LT model (d), $c = 4$ and LT model (e), and $c = 6$ and LT model (f).

Fig. 10. Influence spread computed by the tested algorithms in the grid network: IC model with $d = 0.5$ (a), IC model with $d = 0.7$ (b), IC model with $d = 0.9$ (c), LT model with $d = 0.5$ (d), LT model with $d = 0.7$ (e), and LT model with $d = 0.8$ (f).

Spread approximates the real influence spread function and cannot be compared directly.

The first test is conducted on the Hamstested dataset, both for the IC and the LT models. The simulation shows the performance of our algorithm in terms of influence spread and computational time. In particular, we show in Fig. 11 that the results obtained by the *RIO* algorithm compare the ones obtained by the *Greedy* algorithm while requiring approximately the computational time of the *SelectTopS*.

Figures 11(a) and 11(c) show that the proposed algorithm works with all the seed test cardinality in the, providing high value for the influence spread while employing low computational time. Noted that the time value in Figs. 11(b) and 11(d) are presented on a logarithmic scale; therefore, the gap in the processing times required by the three analyzed algorithms is significantly marked.

Lastly, in Figs. 12(a)-12(d) we report the results of simulation in the case of the ego-Facebook dataset for the IC and

Fig. 11. Comparison of the tested algorithms performance on the Hamsterster dataset: influence spread under the IC model (a), computation time under the IC model (b), influence spread under the LT model (c), and computation time under the LT model (d).

Fig. 12. Comparison of the tested algorithms performance on the ego-Facebook dataset: influence spread under the IC model (a), computation time under the IC model (b), influence spread under the LT model (c), and computation time under the LT model (d).

the LT model. The corresponding results confirm the findings achieved by the previous simulations.

VI. CONCLUSIONS

This paper proposes a novel algorithm for efficiently solving the influence maximization problem in large social networks under the well-known independent cascade (IC) and linear threshold (LT) models. On the one hand, the paper fills a gap in the existing literature, where there is a lack of investigations on low-complexity methods for the computation of the influence spread in the LT model. On the other hand, the work improves the knowledge on approaches that are able to select an initial group of nodes that maximize the spread of influence in large-scale social networks. The complexity analysis and the application to numerical experiments based on real datasets prove that the proposed algorithm ensures to efficiently find near-optimal solutions of the influence maximization problem in a computationally efficient way, both for IC and LT models, showing higher performance in running time and influence spread than recent state of the art approaches.

Future research will address: providing the approximation bound on influence spread, assessing the scalability of the proposed algorithm, modeling uncertainties that may affect the decision parameters.

REFERENCES

- [1] A. Fazeli, A. Ajorlou, and A. Jadbabaie, "Competitive diffusion in social networks: Quality or seeding?" *IEEE Transactions on Control of Network Systems*, vol. 4, no. 3, pp. 665–675, 2016.
- [2] Y. Wang and C. Yu, "Social interaction-based consumer decision-making model in social commerce: The role of word of mouth and observational learning," *International Journal of Information Management*, vol. 37, no. 3, pp. 179–189, 2017.
- [3] F. Bonchi, "Influence propagation in social networks: A data mining perspective," *IEEE Intelligent Informatics Bulletin*, vol. 12, no. 1, pp. 8–16, 2011.
- [4] S. Peng, A. Yang, L. Cao, S. Yu, and D. Xie, "Social influence modeling using information theory in mobile social networks," *Information Sciences*, vol. 379, pp. 146–159, 2017.
- [5] P. Shakaran, A. Bhatnagar, A. Aleali, E. Shaabani, and R. Guo, "The independent cascade and linear threshold models," in *Diffusion in Social Networks*. Springer, 2015, pp. 35–48.
- [6] J. T. Khim, V. Jog, and P.-L. Loh, "Computing and maximizing influence in linear threshold and triggering models," in *Advances in Neural Information Processing Systems*, 2016, pp. 4538–4546.
- [7] F. Lu, W. Zhang, L. Shao, X. Jiang, P. Xu, and H. Jin, "Scalable influence maximization under independent cascade model," *Journal of Network and Computer Applications*, vol. 86, pp. 15–23, 2017.
- [8] D. Liben-Nowell and J. Kleinberg, "Tracing information flow on a global scale using internet chain-letter data," *Proceedings of the national academy of sciences*, vol. 105, no. 12, pp. 4633–4638, 2008.
- [9] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 4, p. 21, 2012.
- [10] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst, "Patterns of cascading behavior in large blog graphs," in *Proceedings of*

- the 2007 SIAM international conference on data mining. SIAM, 2007, pp. 551–556.
- [11] W. Chen, C. Wang, and Y. Wang, “Scalable influence maximization for prevalent viral marketing in large-scale social networks,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 1029–1038.
 - [12] W. Chen, Y. Yuan, and L. Zhang, “Scalable influence maximization in social networks under the linear threshold model,” in *2010 IEEE international conference on data mining*. IEEE, 2010, pp. 88–97.
 - [13] Y. Yang, E. Chen, Q. Liu, B. Xiang, T. Xu, and S. A. Shad, “On approximation of real-world influence spread,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 548–564.
 - [14] E. Mossel and S. Roch, “On the submodularity of influence in social networks,” in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM, 2007, pp. 128–134.
 - [15] C. C. Aggarwal, A. Khan, and X. Yan, “On flow authority discovery in social networks,” in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 522–533.
 - [16] P. Domingos and M. Richardson, “Mining the network value of customers,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 57–66.
 - [17] M. Richardson and P. Domingos, “Mining knowledge-sharing sites for viral marketing,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002, pp. 61–70.
 - [18] D. Kempe, J. Kleinberg, and É. Tardos, “Maximizing the spread of influence through a social network,” in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 137–146.
 - [19] Y. Wang, G. Cong, G. Song, and K. Xie, “Community-based greedy algorithm for mining top-k influential nodes in mobile social networks,” in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 1039–1048.
 - [20] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, “Cost-effective outbreak detection in networks,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 420–429.
 - [21] D. Kempe, J. Kleinberg, and É. Tardos, “Influential nodes in a diffusion model for social networks,” in *International Colloquium on Automata, Languages, and Programming*. Springer, 2005, pp. 1127–1138.
 - [22] S. Zhou, I. J. Cox, and L. K. Hansen, “Second-order assortative mixing in social networks,” in *International Workshop on Complex Networks*. Springer, 2017, pp. 3–15.
 - [23] M.-Y. Zhou, W.-M. Xiong, X.-Y. Wu, Y.-X. Zhang, and H. Liao, “Overlapping influence inspires the selection of multiple spreaders in complex networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 508, pp. 76–83, 2018.
 - [24] L.-I. Ma, C. Ma, H.-F. Zhang, and B.-H. Wang, “Identifying influential spreaders in complex networks based on gravity formula,” *Physica A: Statistical Mechanics and its Applications*, vol. 451, pp. 205–212, 2016.
 - [25] W. Yang, L. Brenner, and A. Giua, “Computation of activation probabilities in the independent cascade model,” in *2018 5th International Conference on Control, Decision and Information Technologies*. IEEE, 2018, pp. 791–797.
 - [26] M. E. Newman, D. J. Watts, and S. H. Strogatz, “Random graph models of social networks,” *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl 1, pp. 2566–2572, 2002.
 - [27] R. Carli and M. Dotoli, “Distributed alternating direction method of multipliers for linearly constrained optimization over a network,” *IEEE Control Systems Letters*, vol. 4, no. 1, pp. 247–252, 2019.
 - [28] W. Yang, L. Brenner, and A. Giua, “Influence maximization in independent cascade networks based on activation probability computation,” *IEEE Access*, vol. 7, pp. 13 745–13 757, 2019.
 - [29] G. Calinescu, C. Chekuri, M. Pal, and J. Vondrák, “Maximizing a monotone submodular function subject to a matroid constraint,” *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.
 - [30] D. Rosa and A. Giua, “On the spread of innovation in social networks,” *IFAC Proceedings Volumes*, vol. 46, no. 27, pp. 322–327, 2013.
 - [31] S. Banerjee, M. Jenamani, and D. K. Pratihar, “A survey on influence maximization in a social network,” *arXiv preprint arXiv:1808.05502*, 2018.
 - [32] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser, “Exceptional paper—location of bank accounts to optimize float: An analytic study of exact and approximate algorithms,” *Management science*, vol. 23, no. 8, pp. 789–810, 1977.
 - [33] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions—i,” *Mathematical programming*, vol. 14, no. 1, pp. 265–294, 1978.
 - [34] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” <http://snap.stanford.edu/data>, Jun. 2014.
 - [35] J. Kunegis, “Konect: the koblenz network collection,” in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 1343–1350.

Paolo Scarabaggio (S'20) received the B.Sc. degree in Mechanical Engineering in 2017 and the M.Sc. degree in Management Engineering in 2019 from the Politecnico di Bari, Bari, Italy. In 2019, he was a visiting student at the Delft Center for Systems and Control, TU Delft, The Netherlands.

He is currently working toward the Ph.D. degree in the Dept. of Electrical and Information Engineering of the same university under the supervision of Prof. Engr. Mariagrazia Dotoli.

His research interests include modeling, optimization, and distributed control of energy and complex networked systems.

Raffaele Carli (M'17) received the Laurea degree in Electronic Engineering with honours in 2002 and the Ph.D. in Electrical and Information Engineering in 2016, both from Politecnico di Bari, Italy. From 2003 to 2004, he was a Reserve Officer with Italian Navy. From 2004 to 2012, he worked as System and Control Engineer and Technical Manager for a space and defense multinational company. Dr. Carli is currently a research fellow at Politecnico di Bari, and his research interests include the formalization, simulation, and implementation of decision and control systems, as well as modeling and optimization of complex systems.

Mariagrazia Dotoli (M'96, SM'12) received the Laurea degree in Electronic Engineering with honours in 1995 and the Ph.D. in electrical engineering in 1999 from Politecnico di Bari (Italy).

She has been a visiting scholar at the Paris 6 University and at the Technical University of Denmark. She is expert evaluator of the European Commission since the 6th Framework Programme. She is a Full Professor in Automatic Control at Politecnico di Bari, which she joined in 1999. She has been the Vice Rector for research

of Politecnico di Bari and a member elect of the Academic Senate. Her research interests include modeling, identification, management, control and diagnosis of discrete event systems, manufacturing systems, logistics systems, traffic networks, smart grids and networked systems.

Prof. Dotoli was co-chairman of the Training and Education Committee of ERUDIT, the European Commission network of excellence for fuzzy logic and uncertainty modeling in information technology, and was key node representative of EUNITE, the European Network of excellence on Intelligent Technologies. She is the Editor in Chief of the IEEE SYSTEMS MAN AND CYBERNETICS SOCIETY E-NEWSLETTER, and an Associate Editor of the IEEE TRANS. ON AUTOMATION SCIENCE AND ENGINEERING, of the IEEE TRANS. ON SYSTEMS, MAN, AND CYBERNETICS and of the IEEE TRANS. ON CONTROL SYSTEMS TECHNOLOGY. She is the Program chair of the 2020 IEEE Conference on Automation Science and Engineering and the General chair of the 29th Mediterranean Conference on Control and Automation. She was the Program co-chair of the 2017 IEEE Conference on Automation Science and Engineering, the Workshop and Tutorial chair of the 2015 IEEE Conference on Automation Science and Engineering, the Special Session co-chair of the 2013 IEEE Conference on Emerging Technology and Factory Automation and chair of the National Committee of the 2009 IFAC Workshop on Dependable Control of Discrete Systems. She was member of the International Program Committee of 80+ international conferences. She is author of 200+ publications, including 1 textbook (in Italian) and 70+ international journal papers.