Enhanced Models for Privacy and Utility in Continuous-Time Diffusion Networks

Federica Granese, Daniele Gorla, Catuscia Palamidessi

Noname manuscript No.
(will be inserted by the editor)

Enhanced Models for Privacy and Utility in Continuous-Time Diffusion Networks

Daniele Gorla · Federica Granese · Catuscia Palamidessi

the date of receipt and acceptance should be inserted later

Abstract Controlling the propagation of information in social networks is a problem of growing importance. On one hand, users wish to freely communicate and interact with their peers. On the other hand, the information they spread can bring to harmful consequences if it falls in the wrong hands. There is therefore a trade-off between utility, i.e., reaching as many intended nodes as possible, and privacy, i.e., avoiding the unintended ones. The problem has attracted the interest of the research community: some models have already been proposed to study how information propagates and to devise policies satisfying the intended privacy and utility requirements. In this paper we adapt the basic framework of Backes et al. to include more realistic features, that in practice influence the way in which information is passed around. More specifically, we consider: (a) the topic of the shared information, (b) the time spent by users to forward information among them and (c) the user social behaviour. For all features, we show a way to reduce our model to the basic one, thus allowing the methods provided in the original paper to cope with our enhanced scenarios. Furthermore, we propose an enhanced formulation of the utility/privacy policies, to maximize the expected number of reached users among the intended ones, while minimizing this number among the unintended ones, and we show how to adapt the basic techniques to these enhanced policies. We conclude by giving a new approach to the

This paper is supported by the ERC project Hypatia under the European Unions Horizon 2020 research and innovation programme. Grant agreement №835294.

D. Gorla
Dept. Computer Science, “Sapienza” Univ. of Rome
E-mail: gorla@di.uniroma1.it

F. Granese
Inria Saclay - cole Polytechnique - IPP - Sapienza, France - Italy
E-mail: federica.granese@inria.fr

C. Palamidessi
Inria Saclay - cole Polytechnique - IPP, France
E-mail: catuscia@lix.polytechnique.fr

maximization/minimization problem by finding a trade-off between the risk and the gain function through bi-objctive optimization.

Keywords Diffusion Networks · Privacy/Utility · Submodular Functions · Regret Ratio.

1 Introduction

In the last decade there has been a tremendous increase in the world-wide diffusion of social networks, leading to a situation in which a large part of the population is highly inter-connected. A consequence of such high connectivity is that, once a user shares a piece of information, it may spread very quickly. The implications of this phenomenon have attracted the attention of many researchers, interested in studying the potentials and the risks behind such implications. The involvement of the scientific community with this topic has already produced a large body of literature; see, for instance, [2,6,8,13,23,31,32], just to cite a few.

In general, *diffusion* [21] is a process by which information, viruses, gossips and any other behaviors spread over networks. Here, we follow a natural and common approach to modeling the net as a graph where nodes represent the users and edges are labeled by the likelihood of transmission between users.

One of the strengths, but also the main potential hazard, of social networks relies on the speed by which information can be diffused: once a piece of information becomes viral, there is no way to control it. This means that it can reach users that it was not meant to reach. If the information is a sensitive one, users naturally have an interest in controlling this phenomenon. In [1], this problem is addressed by defining two types of propagation policies that reconcile privacy (i.e., protecting the information from those who should not receive it) and utility (i.e., sharing the information with those who should receive it). Note that in the framework of [1], instead of considering privacy in terms of an adversary inferring sensitive information from the data published by the user, the authors consider privacy in terms of controlling the spreading of information within a network of users that share the information with each other. Thus the goal is to enable users to share information in social networks in a such a way that, ideally, only the intended recipients receive the information.[1] *Utility-restricted privacy policies* minimize the risk, i.e., the expected number of malicious users that receive the information, while satisfying a constraint on the utility, i.e., a lower bound on the number of friends the user wants to reach. Dually, *privacy-restricted utility policies* maximize the number of friends with whom the information is shared, while respecting an upper bound on the number of malicious nodes reached by the information spread. The authors of [1] prove that *Maximum k-Privacy* - the minimization problem corresponding to the utility-restricted provacy policies - and *Maximum τ-Utility* - the maximization problem corresponding to the privacy-restricted utility policies - are NP-hard, and propose algorithms for approximating the solution.

[1] Even though, this notion of privacy might be known as *confidentiality* or *secrecy*, for the sake of continuity we adopt the terminology as done in [1].

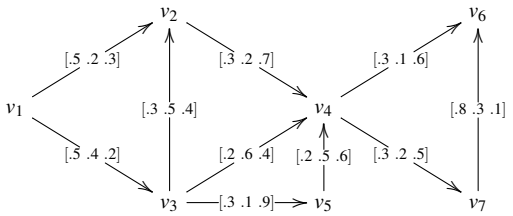


Fig. 1 A Topic vector diffusion network, in which we use topic vectors with three components (*science, movies, society*)

Being one of the first framework to study the trade-off between privacy and utility, the model proposed in [1] is quite basic. One limitation is that the likelihood that governs the transmission along an edge is a constant, fixed in time and irrespective of any other features. We argue that this is not a realistic assumption, and we propose to enrich the framework for modeling the situations described in the following two scenarios.

First, imagine that you are a scientific researcher spending some time on a social network. Suddenly, you see a news about the proof of the century, stating that P = NP. Whom do you wish to share such an information with? Probably with a colleague or someone interested in the subject. To support this kind of scenario, following [14], we consider social networks in which a user may choose the peers to whom to send a piece of information based on the *topic* of that information. To model such a situation, we label the edges of the net by *topic vectors*, defined as vectors in which each component represents the probability of a user to send an information of the corresponding topic (or tag) to the user at the other end of the edge. Furthermore, a piece of information is usually related to several topics, not just one. To model this latter aspect, we also tag a message with a probability distribution (topic distribution) over the topics, representing the weight of each topic in the message. To obtain the probability that a node v_i sends a message to another node v_j, we then consider the scalar product of the topic vector of the edge (v_i, v_j) and the topic distribution of the message.

As an example, assume that there are three topics, *science*, *movies*, and *society*. Figure 1 represents a net whose edges are labeled with instances of these kinds of topic vectors. For example, if v_3 receives a message about a new movie of a director he likes, the probability that he will forward it to v_2 (rather than not) is 0.5, while the probability of forwarding it to v_4 is 0.6 and to v_5 is 0.1, representing the fact that v_2 and v_6 are much more interested than v_5 in the kind of movies that v_3 likes. Note that a topic vector is not a probability vector since the sum of these probabilities is not 1, because these are independent events. Further, consider the P=NP message, and assume that its topic distribution is $(0.9, 0, 0.1)$. Since the edge (v_7, v_6) has topic vector $(0.8, 0.3, 0.1)$, the probability that v_7 sends the message to v_6 is $0.9 \times 0.8 + 0 \times 0.3 + 0.1 \times 0.1 = 0.73$. Note that, being the convex combination of probabilities, the result of such scalar product is always a probability.

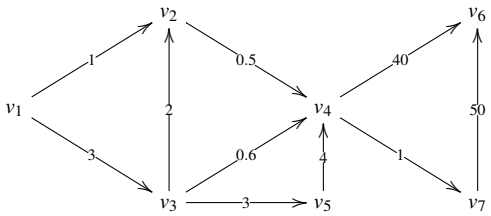


Fig. 2 A Time diffusion network with sampled times for traversing the edge

Second, imagine that you are a night owl; at midnight, you see a funny photo and you want to share it with one of your friends. However, he is a sleepyhead and sleeps all night; thus, he will be able to forward such a photo only the next morning. If we are tracking the diffusion process until a few hours forward, there will be no further diffusion of the photo from your friend. On the other hand, if you had sent the photo during the day, he may have seen and forwarded it soon afterwards. This scenario can be modeled by labeling each edge (v_i, v_j) with a probability density function over time δ_{ij}, representing the probability that the information takes a certain time t for traveling from v_i to v_j. For instance, if v_i is the night owl and v_j the sleepyhead, then it is likely that δ_{ij} will be a big amount of time, but there is still some probability that the information arrives at v_i when they are both awake, in which case the transmission time will be shorter. Each edge may have a different density function: for instance, if v_i has another friend v_z who is a night owl as well, then the moment in which v_z sees the information sent by v_i will be likely to be closer to the one in which v_i forwards the information; hence, the amount of time for the transmission from v_i to v_z will be small. By sampling the time for each edge, we obtain a snapshot of the net, which will have the same structure as a standard net. Figure 2 represents an instance of such a net, where edges labeled with a small value connect users with the same habits. For example, v_2 and v_4 can be both night owls, whereas v_6 is a sleepyhead (or vice versa).

A third variant of the basic framework can again be derived from the example just examined. Indeed, if you are a night owl, then probably you will tend to send a funny photo during the night instead that during the morning. From this toy example, we can deduce that the social behaviour of the users deeply influences the information diffusion. To model this scenario, we suppose to have some information on the habits of users in a certain time interval (like a week, a month, a year). For example we might know when a user usually signs-in on a specific social network during a week. Starting from this knowledge, we associate each node u with a continuos distribution (*concrete time model*) or with a set of formulas (*logical time model*) encoding the behaviour of u in any moment of the week. For instance, if the traffic data of u is thickened on friday nights, then u will be equipped with a probability distribution

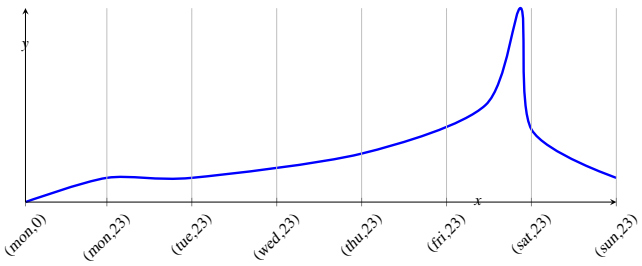


Fig. 3 Week distribution of u.

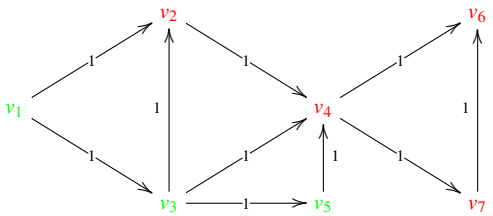


Fig. 4 A General diffusion network in which green nodes are friends and red nodes are malicious.

like the one in Figure 3 or with the following set of formulas:

$$\begin{aligned}\Phi_u = \{&S = (fri, AM) : 0.3,\\ &S = (fri, PM) : 0.6,\\ &\neg(S = (fri, AM) \vee S = (fri, PM)) : 0.1\}.\end{aligned}$$

where a formula "$S = \Sigma : p$" should be read as: "in the time slot Σ, the probability for u to connect is p" (here, S denotes the current time moment). Starting from these probabilities, we can recover the probability of communication between two users during a time laps by constructing a Markov Chain.

Another limitation of the standard framework is in the way in which the trade-off problem is formulated in [1]: for maximizing privacy and utility, the corresponding problems try to minimize the number of malicious nodes infected *up to time t* (given a bound on the number of friends *initially* sharing the information), or to maximize the number of friends *initially* sharing the information (given a bound on the number of malicious nodes infected *up to time t*). By contrast, we argue that utility would be better expressed in terms of the friends reached by the information *up to time t*, instead of the initial friends only. Furthermore, privacy and utility would be more symmetric, in that both of them would be expressed in terms of nodes reached at time t.

As an example, consider Figure 4 and suppose we want to monitor the diffusion up to time $t = 1$. Consider first the maximum utility problem under the constraint of reaching (at time $t = 1$) at most one malicious node. In the standard framework, there are two solutions for the set of initial nodes: either $\{v_1\}$ or $\{v_5\}$. They are considered

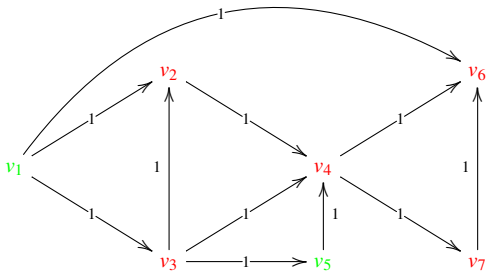


Fig. 5 Another General diffusion network in which green nodes are friends and red nodes are malicious.

equivalent because we only consider further infection of the malicious nodes (and in both cases, in 1 time unit just one malicious node gets infected). By contrast, we argue that $\{v_1\}$ is a better solution, because if we start with $\{v_1\}$ then in 1 time unit the information will reach also the friend node v_3, whereas no further friend will be reached if we start with $\{v_5\}$.

Consider now the maximum privacy problem. Assume that we want to minimize the number of malicious nodes infected up to time $t = 1$ under the constraint of having at least two friends sharing the information. The solution of the problem in [1] is any subset formed by two friend nodes. Any such subset, in fact, leads to infect two malicious nodes at time $t = 1$. By contrast, we argue that the optimal solution would be the (smaller) initial set $\{v_1\}$. In fact, this solution would respect the constraint if, as we propose, we count also the friends infected at time $t = 1$, and would minimize the malicious nodes infected in the same time unit.

The *fil rouge* of the before scenarios is the presence of constraints on utility or privacy. Suppose we want to overcome this scheme. A way is to find a trade-off between the solutions maximizing the infection between the friend nodes and minimizing the infection between the malicious ones. As an example, consider Figure 5 and suppose we want to monitor the diffusion up to time $t = 2$. As we can see, v_1 optimizes utility but not privacy, whereas v_5 does the opposite: v_1 infects 1 friend and 4 malicious nodes, whilst v_5 infects 0 friends and 3 malicious nodes. In this case, we can calculate the so called *non-dominated solutions* and leave the user decide according to him/her preference. In this toy example, the possible solutions are the subsets of $\{v_1, v_5\}$ and the non-dominated solutions are $\{v_1\}$ and $\{v_5\}$. If the user is more concerned on utility than on privacy, then he will initially share the information with v_1; otherwise, he will send the information to v_5.

In practice, enumerating all these solutions is a difficult task since they can be exponentially many. For this reason, we show to the user just a subset of them. Let us denote by $\mathscr{C}$ the whole set and by $\mathscr{S} \subseteq \mathscr{C}$ the subset returned. We want $\mathscr{S}$ be as much representative of $\mathscr{C}$ as possible. We approach this problem by the notion of *regret ratio*, that measures the degree of satisfaction of the user in choosing from $\mathscr{S}$ instead of $\mathscr{C}$.

Contributions To sum up, the contributions of this paper are the following:

- We extend the basic graph diffusion model proposed in [1] by considering a more sophisticate labeling of the edges. This allows us to take into account, for the propagation of information, (a) the topics; (b) the probabilistic nature of the transmission rates; and (c) the user social behaviours.
- We reformulate the optimization goals of [1] by considering a notion of utility which takes into account the friend nodes reached up to a certain time t, rather than the initial set only. We argue that this notion is more natural, besides being more in line with that of privacy (the infected malicious nodes are counted up to time t as well).
- We prove that the resulting optimization problems are NP-hard and provide suitable approximation algorithms. Moreover, we want to stress we conserve the same approximation ratio as the basic framework.
- We provide a new formulation of the problem by considering utility and privacy as the objectives of a multiobjective problem. We also solve the problem by approximating the Pareto front.

Related Work There is a huge literature on information propagation in social networks, but most of the papers focus on maximizing the spread of information in the whole network. See for instance [7,17,18,22,27]. To make such works closer to real life situations, some papers revisit them on either the influence problem or the network model. For example, in [4,5,30], the problem is modified by considering the scenario where a company wants to use viral marketing to introduce a new product into a market when a competing product is simultaneously being introduced. Referring to A and B as the two technologies of interest, they denote with I_A (I_B) the initial set of users adopting technology A (B). Hence, they try to maximize the expected number of consumers that will adopt technology A, given I_A and I_B, under the assumption that consumers will use only one of the two products and will influence their friends on the product to use. In [4], the authors consider the problem of limiting the spread of misinformation in social networks. Considering the setting described before (with the two competitive companies), they refer to one of the two companies as the "bad" company and to the other one as the "good" company.

In the papers mentioned so far, authors always assume that all the selected top influential nodes propagate influence as expected. However, some of the selected nodes could not work well in practice, leading to influence loss. Thus, the objective of [34] is to find the set K of the most influential nodes with which initially the information should be shared, given a threshold on influence loss due to a failure of a subset of nodes $R \subseteq K$. This problem, as all the previous ones, are proven to be NP-hard; furthermore, all of [4,5,30,34] assume that the diffusion process is timeless.

A different research line consists in making the underlying network model closer to reality, instead of modifying the problem itself. For example, topic of information is handled in [14], where the authors infer what we call topic vector. Always considering the information item, the model in [33] endows each node with an influence vector (how authoritative they are on each topic) and a receptivity vector (how susceptible they are on each topic). While for diffusion networks there exists a good amount literature about the role of users' interests [14,33,35,36], the same is not true for the role of time with respect to user habits.

An orthogonal research line is represented by works like [14,16], aiming at inferring transmission likelihoods: given the observed infection times of nodes, they infer the edges of the global diffusion network and estimate the transmission rates of each edge that best explain the observed data. This leads to an interesting problem that can be solved with convex optimization techniques. Note that, as in [1], we are not dealing with this aspect, since we assume that the inference has already happened and we have an accurate estimate of the transmission likelihoods (whatever they are) for the whole network.

Paper Organization This paper is organized as follows. In Section 2, we recall the basic notions and results from [1] and the basic ingredients for our optimization problem. Then, in Section 3, we present the enhanced models, one where information transmission is ruled by the topic of conversation, another one based on the transmission time and the last one based on user habits on social networks. In Section 4.1, we modify the basic definitions of utility-restricted privacy policies and privacy-restricted utility policies, and show that all the theory developed in [1] with the original definitions can be smoothly adapted to these new (and more realistic) definitions. In Section 4.2 we provide a new formulation of the problem by considering utility and privacy as the objectives of a multiobjective problem. Finally, in Section 5, we conclude the paper, by also drawing lines for future research. A preliminary version of this paper appeared in [19]. In the present version we have improved the presentation, and added new material, including:

1. two new models of diffusion networks, namely *Concrete Time Diffusion Network* and *Logic Time Diffusion Network*, which allow us to study how users' social behaviors influence the diffusion of information on a network;
2. we have considered the *Gain-Risk Maximization Problem*, and we have solved it by using the approach of *Regret Ratio*.

2 Background

In this section we recall the basic notions from [1] and from [12], which will be used in the rest of the paper.

2.1 Submodular Functions

Definition 1 (Submodular function [15]) A function $f\colon 2^V \to \mathbb{R}$ is *submodular* if, for all $S,T \subseteq V$, it holds that $f(S)+f(T) \geq f(S\cup T)+f(S\cap T)$.

Let $f(j|S) := f(S\cup\{j\}) - f(S)$ be the the *profit* (or *cost*) of $j \in V$ in the context of $S \subseteq V$; then, it is known that f is submodular iff $f(j|S) \geq f(j|T)$, for all $S \subseteq T$ and $j \notin T$. The function f is *monotone* iff $f(j|S) \geq 0$, for all $S \subseteq V$ and $j \notin S$.

Given a submodular function f, the *curvature* κ_f of f is defined as

$$\kappa_f := min_{j\in V} \frac{f(j|V\setminus\{j\})}{f(\{j\})}$$

Intuitively, this factor measures how close a submodular function is to a modular function, where $f\colon 2^V \to \mathbb{R}$ is *modular* if, for all $S,T \subseteq V$, $f(S)+f(T) = f(S\cup T)+f(S\cap T)$. The closer the function to being modular i.e., the smaller the curvature), the easier it is to optimize.

Remark 1 It is well known that the linear combination of submodular functions is still submodular.

2.2 Diffusion Networks

Definition 2 (General Diffusion Network) A *general diffusion network* is a tuple $N = (V,\gamma)$, where $V = \{v_i\}_{i=1\ldots n}$ is the set of nodes and $\gamma = (\gamma_{ij})_{i,j=1\ldots n}$ is the transmission matrix of the network (with $\gamma_{ij} \geq 0$, for all i,j).

Thus, V and γ define a directed graph where each $\gamma_{ij} > 0$ represents an edge between nodes v_i and v_j along which the information can potentially flow, together with the flow likelihood. Let us now consider a general diffusion network N in which $F \subseteq V$ is the set of friendly nodes and $M \subseteq V$ is the set of malicious nodes, with $F \cap M = \emptyset$. The idea is to maximize the number of friends and minimizing the number of enemies reached by an information in a certain time window.

Definition 3 (Utility-restricted Privacy Policy) A *utility-restricted privacy policy* Π is a 4-tuple $\Pi = (F,M,k,t)$ where F is the set of friend nodes, M is the set of malicious nodes, k is the number of nodes the information should be shared to, and t is the period of time in which the policy should be valid.

Definition 4 (Privacy-restricted Utility Policy) A *privacy-restricted utility policy* Υ is a 4-tuple $\Upsilon = (F,M,\tau,t)$ where F is the set of friend nodes, M is the set of malicious nodes, τ is the expected number of nodes in M receiving the information during the diffusion process, and t is the period of time in which the policy should be valid.

Both the policies are focused on bounding the risk that a malicious node gets infected by time t, given that $F' \subseteq F$ is initially infected.

Definition 5 (Risk) Let N be a diffusion network. The *risk* $\rho_N(F',M,t)$ *caused by* $F' \subseteq V$ *with respect to* $M \subseteq V$ *within time* t is given by

$$\rho_N(F',M,t) = \sum_{m\in M} Pr[t_m \leq t|F']$$

Here, $Pr[t_m \leq t|F']$ is the likelihood that the infection time t_m of malicious node m is at most t, given that F' is infected at time $t = 0$.

Hence, the risk function gives us an upper bound on the number of malicious nodes receiving the information in a given time window, given that a subset of friendly nodes was initially infected. This definition of risk function recalls the one of *influence function* [1,17,18]. Here, instead of being interested in the expected number of infected nodes in a set of malicious nodes, we are interested in the infection

in the whole network. Thus, given $A \subseteq V$, the influence in the network N within time t is denoted by $\sigma_N(A,t)$. In [18] it is shown that computing $\sigma_N(A,t)$ is #P-hard and they approach the problem of the influence estimation by using a randomized approximation algorithm. As already written in [1], since the risk function is just a generalization the regular influence function, computing $\rho_N(F',M,t)$ is also #P-hard; however, we can use the algorithm in [18] to approximate the risk function up to a constant factor: we simply ignore the infection times for nodes not in M.

To make notation lighter, we shall usually omit the subscript N from ρ_N, when clear from the context. To maximally satisfy a utility-restricted privacy policy and a privacy-restricted utility policy, the following two problems are defined.

Definition 6 (Maximum k-privacy – MP) Given a utility-restricted privacy policy $\Pi = (F,M,k,t)$ and a general diffusion network N, the *maximum k-privacy problem* (MP, for short) is given by

$$\begin{array}{ll} \underset{F' \subseteq F}{\text{minimize}} & \rho(F',M,t) \\ \text{subject to} & |F'| \geq k \end{array} \tag{1}$$

Definition 7 (Maximum τ-utility – MU) Given a privacy-restricted utility policy $\Upsilon = (F,M,\tau,t)$ and a general diffusion network N, the *maximum τ-utility problem* (MU, for short) is given by

$$\begin{array}{ll} \underset{F' \subseteq F}{\text{maximize}} & |F'| \\ \text{subject to} & \rho(F',M,t) \leq \tau \end{array} \tag{2}$$

The idea behind MP is to look for a subset of at least k friendly nodes with which initially share the information, in order to minimize the diffusion between malicious nodes at time t. By contrast, MU looks for the maximum set of friendly nodes with which initially share the information, in order to infect at most τ malicious nodes at time t. Both problems are NP-hard. However, they can be approximated and the approximation algorithms rely on the submodularity of the risk function: since ρ is submodular, monotone and with a non-zero curvature, it is possible to derive an efficient constant factor approximation, where the approximation factor depends on the structure of the underlying network N.

Optimizing submodular functions is a difficult task, but we can get around the problem by choosing a proper surrogate function for the objective and optimize it; the surrogate functions usually are upper or lower bounds. For example, the *majorization-minimization* algorithms begin with an arbitrary solution Y to the optimization problem and then optimize a modular approximation formed via the current solution Y. Therefore, following the work in [1,20], we can solve MP (and MU) by choosing a surrogate function for ρ. In Algorithm 1, given a candidate solution $Y \subseteq F$, the modular approximation of the risk function ρ is given by

$$m_{g_Y}(X) = \rho(Y) + g_Y(X) - g_Y(Y)$$

where

$$g_Y(X) = \sum_{v \in X} g_Y(v)$$

and

$$g_Y(v) = \begin{cases} \rho(v|F \setminus \{v\}), & \text{if } v \in Y \\ \rho(v|Y), & \text{otherwise.} \end{cases}$$

Due to the submodularity of the risk function, we can use this submodular approximation as an upper bound for the risk, i.e. $m_{g_Y}(Y) \geq \rho(Y)$ [1,20].

Algorithm 1 Maximum k-Privacy

Require: Instance F, M, k of maximum k-privacy
Ensure: $satisfyingMP(F,M,k)$
1: $C \leftarrow \{X \subseteq F \,:\, |X| = k\}$
2: Select a random candidate solution $X^1 \in C$
3: $t \leftarrow 0$
4: **repeat**
5: $\quad t \leftarrow t+1$
6: $\quad X^{t+1} \leftarrow argmin_{X \in C}\, m_{g_{X^t}}(X)$
7: **until** $X^{t+1} = X^t$
8: **return** X^t

At each iteration, Algorithm 1 finds the new set that minimizes the upper bound of the risk function. Clearly, since this set minimizes the upper bound of the risk function, it also minimizes the risk function.[2] Now, recall that the curvature $\kappa_{\rho(F,M,t)}$ of $\rho(F,M,t)$ is given by

$$\kappa_{\rho(F,M,t)} := min_{v \in F} \frac{\rho(v|F \setminus \{v\},M,t)}{\rho(\{v\},M,t)}$$

where $\rho(v|F \setminus \{v\},M,t) := \rho(F,M,t) - \rho(F \setminus \{v\},M,t)$. This quantity can be used to give the approximation factor.

Theorem 1 *Algorithm 1 approximates maximum k-privacy to a factor* $\frac{1}{\kappa_\rho}$. *That is, let* F' *be the output and* F^* *be the optimal solution; then,* $\rho(F',M,t) \leq \frac{1}{\kappa_\rho}\rho(F^*,M,t)$.

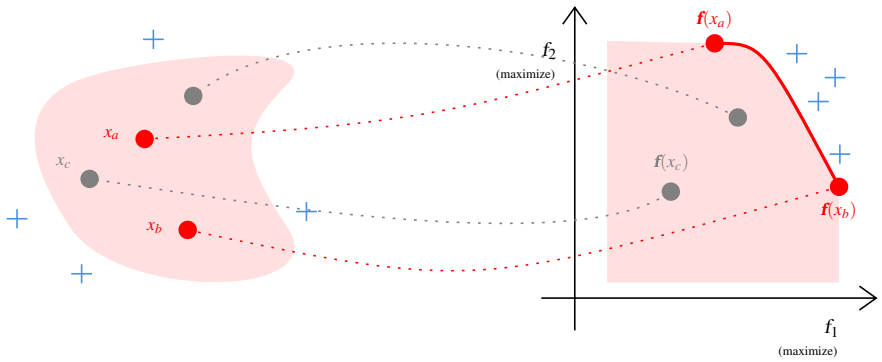
Starting from the approximation algorithm for maximum k-privacy, maximum τ-utility can be approximated through Algorithm 2.

Theorem 2 *Let* n^* *be the optimal solution to an instance of maximum* τ*-utility, and let n be the output of Algorithm 2 for the same instance, using a* $\frac{1}{\kappa_\rho}$*-approximation for maximum k-privacy. Then* $n \geq \kappa_\rho n^*$.

[2] This methodology can be seen as the gradient descent method for minimizing continuous differentiable functions: we start from a random point y and we iteratively move in the direction of the steepest descent, as defined by the negative of the gradient.

Algorithm 2 Maximum τ-Utility

Require: Instance F, M, τ of maximum τ-utility
Ensure: $satisfyingMU(F,M,\tau)$
1: **for** $n \in [|F|,\dots,1]$ **do**
2: $\quad \tau' \leftarrow min_{F' \subseteq F}\ \rho(F',M,t)$ s.t. $|F'| = n$
3: $\quad$ **if** $\tau' \leq \tau$ **then**
4: $\quad\quad$ **return** n
5: **return** 0



(a) Decision Space (b) Objective Space

Fig. 6 MOP with two objective functions to maximize. The figure shows a mapping from the decision space to the objective space. The red curve is the *Pareto Front*; the points in red are examples of *non-dominated* solutions; the points in gray are examples of *dominated* solutions; the points in blue are examples of *infeasible* solutions.

2.3 Background on Multiobjective Optimization

Consider a typical design problem with more than one objective. Given m objective functions $f_1 \dots f_m : \mathscr{X} \to \mathbb{R}$, $\mathscr{X}$ is called decision space and it is the set of all feasible solutions. The set $\mathbb{R}^m$ is called objective space and it contains the evaluations of the solutions through the objective functions, i.e. putting $\boldsymbol{f} = (f_1,\dots,f_m)^\top$ then $\forall x \in \mathscr{X}$ there exists $\boldsymbol{f}(x) \in \mathbb{R}^m$. Let us consider the example in Figure 6. The decision space $\mathscr{X}$ (pink area of Figure 6(a)) is mapped in the objective space, subset of $\mathbb{R}^2$ (pink area of Figure 6(b)).

A multiobjective optimization problem (MOP) is given by the following problem statement:

$$\underset{x \in \mathscr{X}}{\text{minimize/maximize}} \quad f_1(x),\dots,f_m(x).$$

For the sake of simplicity, we shall consider m objective functions to be simultaneously maximized, since this is equivalent to having both maximization and minimization (it is easy to transform a maximization problem in a minimization one and viceversa – see, e.g., Lemma 1).

The difficulty in multiobjective optimization arises from the fact that there may be no one single solution that maximizes all the objective functions simultaneously. Hence, in MOP we want to find the set of *trade-offs* solutions, i.e. solutions that can

improve one objective without deteriorating the performance in any other objective. For doing that, we need a measure with which we can compare the solutions. To this aim, we introduce the notion of *Pareto dominance*.

Definition 8 (Pareto Dominance, ε-Pareto Dominance) Given $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{R}^m$, the point $\boldsymbol{y}_1$ is said to *Pareto dominate* the point $\boldsymbol{y}_2$ (written $\boldsymbol{y}_1 \geq_P \boldsymbol{y}_2$) if and only if $\forall i \in \{1,\ldots,m\} : y_{1i} \geq y_{2i}$ and $\exists j \in \{1,\ldots,m\} : y_{1j} > y_{2j}$. Given $\varepsilon \geq 0$, we say that $\boldsymbol{y}_1$ *Pareto ε-dominates* $\boldsymbol{y}_2$ (written $\boldsymbol{y}_1 \geq_{P,\varepsilon} \boldsymbol{y}_2$) if $\boldsymbol{y}_1 \geq_P (1+\varepsilon)\boldsymbol{y}_2$. Accordingly, we say that $x_1 \in \mathscr{X}$ *dominates* (or *ε-dominates*) $x_2 \in \mathscr{X}$ if $\boldsymbol{f}(x_1) \geq_P \boldsymbol{f}(x_2)$ (or $\boldsymbol{f}(x_1) \geq_{P,\varepsilon} \boldsymbol{f}(x_2)$).

Notationally, when clear from the context, we drop P from $\geq_P$ and $\geq_{P,\varepsilon}$. For instance, in Figure 6 $\boldsymbol{f}(x_c)$ is dominated by both $\boldsymbol{f}(x_a)$ and $\boldsymbol{f}(x_b)$, since $f_1(x_c) < f_1(x_a)$, $f_2(x_c) < f_2(x_a)$, $f_1(x_c) < f_1(x_b)$ and $f_2(x_c) < f_2(x_b)$. Moreover, neither of $\boldsymbol{f}(x_a)$ and $\boldsymbol{f}(x_b)$ dominates the other since $f_2(x_a) > f_2(x_b)$ but $f_1(x_a) < f_1(x_b)$.

The evaluations in the objective space of the non-dominated solutions form the so called *Pareto front*.

Definition 9 (Pareto Optimum, Pareto Set, Pareto Front) Given $x^* \in \mathscr{X}$, then x^* is a *Pareto optimum* if there is no $x \in \mathscr{X}$ s.t. x dominates x^*. The set of Pareto optimum solutions form the *Pareto set*. The *Pareto front* (or *Pareto curve*) is the set formed by the evaluations of the Pareto optimum solutions in the objective space, i.e. $Par(\mathscr{X}) = \{\boldsymbol{f}(x^*) | x^* \in \mathscr{X}, x^* \text{ is Pareto optimum}\}$.

Let us consider again Figure 6; x_a and x_b are two Pareto optima and their evaluations in the objective space belong to the Pareto front. By following Definition 9, we can define the Pareto optimum, Pareto set and Pareto front in the case of ε approximation.

Definition 10 (ε-Pareto Optimum, ε-Pareto Set, ε-Pareto Front) Given $x^* \in \mathscr{X}$, then x^* is a *ε-Pareto optimum* if there is no $x \in \mathscr{X}$ s.t. x ε-dominates x^*. The set of ε-Pareto optimum solutions form the *ε-Pareto set*. The *ε-Pareto front* (or *ε-Pareto curve*) is the set formed by the evaluations of the ε-Pareto optimum solutions in the objective space.

Note that every instance of a multiobejctive problem has a unique Pareto set but in general it has many different ε-Pareto sets of drastically different size. It is known that, for every multiobjective problem with a fixed number m of polynomially computable objective functions, there exists an ε-Pareto set of polynomial size, in particular of size $O((\frac{n}{\varepsilon})^{m-1})$, where n is the bit complexity of the objective functions [12]. Necessary and sufficient conditions for polynomial time constructibility of ε-Pareto sets are given in [11,25].

In a multiobjectve problem we would ideally like to compute the Pareto front. The problem is that the Pareto curve has typically an enormous number of points and thus we cannot compute them. We can only compute a limited number of solutions that optimally approximate the Pareto curve; thus, we want to find a set of solutions as close as possible to the Pareto front and such that the solutions are as diverse as possible. In Section 4.2, we shall present a possible way to reach this goal in our

framework, by adapting the technique developed in [29], relying on the notion of *regret ratio*. This notion was introduced in [24] for obtaining a subset of representative points from a point set; the maximum regret ratio measured how bad the users could feel if they have to choose from the returned points instead of from the entire set.

3 Enhanced Models

In this section, we provide three different models which enhance general diffusion networks by including more realistic and informative features; this is done mainly by using different kinds of transmission matrices. In particular, in the first model, called *topic vector diffusion network*, we bind the likelihood of transmitting an information to the topic of that information. In the second model, called *time diffusion network*, we bind the likelihood to the amount of time an information takes for been transmitted. Finally, in the third model, called *absolute time diffusion network*, transmissions are governed by the user habits within a certain time frame, e.g. a week; according to the information available (whether on a day-by-day base or on a more abstract one, given through logical formulae), we further distinguish the model into the *concrete time* and the *logical time* one. In all scenarios, by following [1], we are not interested in the inference of the transmission likelihoods, and our goal will be the reduction of the enhanced models to the original one, for which the two kinds of policies (MP and MU) are defined.

3.1 Topic Vector Diffusion Networks

We first consider a social network where edges are labeled by *topic vectors*, that are vectors in which each component represents the probability of a user to send an information of the corresponding *topic* (or *tag*) to another user.

Definition 11 (Topic Vector Diffusion Network) A *topic vector diffusion network* is a tuple $N_{TV} = (V, \mathbf{A}, k)$, where $V = \{v_i\}_{i=1\ldots n}$ is the set of nodes in the network, k is the number of topics and $\mathbf{A} = (\boldsymbol{\alpha_1}, \ldots, \boldsymbol{\alpha_n})$ is s.t. $\boldsymbol{\alpha_i}$ is the matrix of dimension $n \times k$ giving the topic vector that rules the transmission rates from node v_i to the other nodes in the network. That is,

$$\boldsymbol{\alpha_i} := \begin{pmatrix} \alpha_{i1}^1 & \ldots & \alpha_{i1}^k \\ \vdots & & \vdots \\ \alpha_{in}^1 & \ldots & \alpha_{in}^k \end{pmatrix}$$

where every $\boldsymbol{\alpha_{ij}} = (\alpha_{ij}^1 \ldots \alpha_{ij}^k)$ is called *topic vector* and each α_{ij}^l (for $l = 1 \ldots k$) is the probability that user i sends an information of topic l to user j.

Notice that a topic vector is not required to be a probability distribution and that, for every i, j and l, the probability of not sending an information of topic l from i to j is $1 - \alpha_{ij}^l$. Together, V and $\mathbf{A}$ define a weighted directed graph where each $\boldsymbol{\alpha_{ij}}$ (i.e., each row of $\boldsymbol{\alpha_i}$ having non zero components) represents an edge between v_i

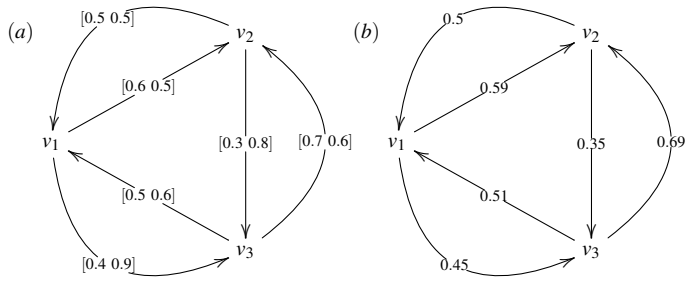


Fig. 7 From a topic vector diffusion network to the $\boldsymbol{m}$-diffusion network. (a) A topic vector diffusion network. (b) The associated $(0.9\ 0.1)$-Diffusion network

and v_j with weight $\boldsymbol{\alpha_{ij}}$. For example, consider the network N_{TV} in Figure 7(a), with $V = \{v_1, v_2, v_3\}$, $k = 2$ and

$$\boldsymbol{\alpha_1} = \begin{pmatrix} 0 & 0 \\ 0.6 & 0.5 \\ 0.4 & 0.9 \end{pmatrix}, \ \boldsymbol{\alpha_2} = \begin{pmatrix} 0.5 & 0.5 \\ 0 & 0 \\ 0.3 & 0.8 \end{pmatrix}, \ \boldsymbol{\alpha_3} = \begin{pmatrix} 0.5 & 0.6 \\ 0.7 & 0.6 \\ 0 & 0 \end{pmatrix}$$

Then, user v_1 sends to v_2 an information about topic 1 with probability 0.6 and an information about topic 2 with probability 0.5.

Definition 12 (Information Item) An *information item* (or *meme*) is a k-dimensional probability vector, in which each component is the weight of a topic relating to the subject of the information. That is, $\boldsymbol{m} := (m_1 \dots m_k)$ such that $m_1 + \dots + m_k = 1$.

For instance, consider vectors consisting of two components, *science* and *society*. The information item associated to a tweet on a scientific paper could be $\boldsymbol{m} = (0.9\ 0.1)$.

Remark 2 A topic vector is different from a meme since it is not a probability vector (indeed, each component of a topic vector is itself a probability).

Definition 13 (Probability of Infection of an Information Item) Let N_{TV} be a topic vector diffusion network, $i, j \in V$ and $\boldsymbol{m}$ the input meme. Then, the probability that i sends $\boldsymbol{m}$ to j is given by:

$$\beta_{ij\boldsymbol{m}} = \boldsymbol{\alpha_{ij}}\boldsymbol{m}^\top \tag{3}$$

Notice that, since each component of $\boldsymbol{\alpha_{ij}}$ is a probability and $\boldsymbol{m}$ is a probability vector, we obtain:

$$0 = \mathbf{0}\boldsymbol{m}^\top \leq \boldsymbol{\alpha_{ij}}\boldsymbol{m}^\top \leq \mathbf{1}\boldsymbol{m}^\top = 1$$

Definition 14 ($\boldsymbol{m}$-Diffusion Network) An *$\boldsymbol{m}$-diffusion network* is a tuple $N_m = (V, \beta_{\boldsymbol{m}})$, where $V = \{v_i\}_{i=1\dots n}$ is the set of nodes and $\boldsymbol{\beta_m} = (\beta_{ij\boldsymbol{m}})_{i,j=1\dots n}$ is the transmission matrix of the network that forwards $\boldsymbol{m}$ (with $\beta_{ij\boldsymbol{m}} \geq 0$ for every i, j).

Given a topic vector diffusion network and an information item, we can derive the associated $\boldsymbol{m}$-diffusion network by determining the probability of infection between each node with respect to the information item (i.e., the transmission matrix $\boldsymbol{\beta}_{\boldsymbol{m}}$). Resuming the example before, with $\boldsymbol{m} = (0.9\ 0.1)$ representing the information item of a scientific paper, consider the topic diffusion network in Figure 7(b), in which we suppose the topic vectors have the same tags as $\boldsymbol{m}$ (science and society). By Definition 13, we have, e.g., that $\beta_{32\boldsymbol{m}} = (0.7\ 0.6)(0.9\ 0.1)^\top = 0.69$ and $\beta_{31\boldsymbol{m}} = (0.5\ 0.6)(0.9\ 0.1)^\top = 0.51$; hence, the probability that v_3 forwards $\boldsymbol{m}$ to v_2 is greater than the probability of forwarding it to v_1, since $\boldsymbol{m}$ is more focused on science than on society.

Even if $\boldsymbol{m}$-diffusion networks seem similar to general diffusion networks, they still have an important difference: in them, transmission depends on the information item. Thus, modeling real-life networks is more natural and more accurate with this enhanced framework. Nonetheless, the new framework can be reduced to the basic one via the transformation we are going to describe now. This of course has the advantage of reusing all the theory developed in [1] for free. To this aim, consider a sample of memes $T = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_h\}$ and their associated $\boldsymbol{m}_l$-diffusion networks derived from the same topic vector diffusion network. Let us concentrate on two nodes i, j in V and define the independent events $E_{ijl} = \{i \textit{ sends } \boldsymbol{m}_l \textit{ to } j\}$; clearly, $Pr(E_{ijl}) = \beta_{ij\boldsymbol{m}_l}$. We can define a random variable X_{ij} counting the number of information items in T sent from i to j. Thus, we can compute the probability that i sends $0, 1, \ldots, h$ information items to j as follows:

$$
\begin{aligned}
Pr(X_{ij} = 0) &= \prod_{l=1}^{h} (1 - \beta_{ij\boldsymbol{m}_l}) \\
&\vdots \\
Pr(X_{ij} = d) &= \sum_{\{l_1,\ldots,l_d\} \subseteq \{1,\ldots,h\}} \beta_{ij\boldsymbol{m}_{l_1}} \ldots \beta_{ij\boldsymbol{m}_{l_d}} \left(\prod_{l' \in \{1,\ldots,h\} \setminus \{l_1,\ldots,l_d\}} (1 - \beta_{ij\boldsymbol{m}_{l'}}) \right) \\
&\vdots \\
Pr(X_{ij} = h) &= \prod_{l=1}^{h} \beta_{ij\boldsymbol{m}_l}
\end{aligned}
$$

The derivation of the general diffusion network from a set of $\boldsymbol{m}_l$-diffusion networks (obtained from the same topic vector diffusion network) is given by first computing for each $i, j \in V$

$$\mathbb{E}[X_{ij}] = \sum_{d=1}^{h} d Pr\,(X_{ij} = d)$$

Then, by starting from these expected values and by dividing by h (for the sake of normalization), we can recover a general diffusion network: the set of nodes remains V and $\gamma_{ij} := \frac{\mathbb{E}[X_{ij}]}{h}$, for every i, j.

3.2 Time Diffusion Networks

We now consider a diffusion network in which each edge (v_i, v_j) is equipped with a probability density function describing, for any given time interval (providing the time spent by the information in traveling along it), the probability of transmitting along that edge.

Definition 15 (Time transmission function) A *time transmission function* $f(\delta)$ is a a probability density function over a time interval.

Definition 16 (Time diffusion network) A *time diffusion network* is a tuple $N_T = (V, \zeta)$, where $V = \{v_i\}_{i=1...n}$ is the set of nodes in the network and $\zeta = (f_{ij}(\delta_{ij}))_{i,j=1...n}$ is the transmission matrix of the network, with $f_{ij}(\cdot)$ a time transmission function and δ_{ij} a time interval (for every i and j).

In contrast with the discrete-time model (which associates each edge with a fixed infection probability), this model associates each edge with a probability density function. Moreover, instead of considering parametric transmission functions such as exponential distribution, Pareto distribution or Rayleigh distribution, we consider the non-parametric ones because in real word scenarios the waiting times obey to different distributions. So, for example, if two nodes are usually logged-in simultaneously (hence, their respective delay in transmission is small), the time function will assign high probabilities to short intervals and negligible probabilities to long ones; the situation is dual for users that are usually logged-in during different moments of the day.

Now suppose that some external agent gives in input to some nodes of the network a certain information at time $t = 0$. Each of these nodes try to forward this information to their neighbors; clearly, this entails a certain amount of time.

Definition 17 (Transmission time) Given two neighbor nodes i and j of a time diffusion network, the *transmission time* δ_{ij} is the amount of time the information requires for going from i to j during a diffusion process.

Starting from a time diffusion network N_T, we can compute the random transmission times associated to each edge of the network by drawing them from the corresponding transmission functions. Consider now a diffusion process over a time diffusion network N_T with the sampled transmission times and suppose that the initial set of infected nodes is F'.

Definition 18 (Infection time of a node [18]) The *infection time* of $v \in V$ is given by:

$$t_v(\{\delta_{ij}\}_{(i,j)\in N_T} | F') := min_{q\in Q_v(F')} \sum_{(i,j)\in q} \delta_{ij}$$

where F' is the set of nodes infected at time $t = 0$ and $Q_v(F')$ is the set of the directed paths from F' to v.

For preserving Theorems 1 and 2 also in this setting, we must first prove submodularity of the risk function on time diffusion networks. For this purpose, let us slightly modify Definition 5.

Definition 19 (Risk) Let $N_T = (V, \zeta)$ be a time diffusion network. The *risk* $\rho_{N_T}(F', M, t)$ caused by $F' \subseteq V$ with respect to $M \subseteq V$ within time t is given by

$$\rho(F', M, t) := \sum_{m \in M} Pr[t_m(\{\delta_{ij}\}_{(i,j)\in N_T} | F') \leq t].$$

Here, $Pr[t_m(\{\delta_{ij}\}_{(i,j)\in N_T} | F') \leq t]$ is the likelihood that the infection time t_m of malicious node m is at most t, given that F' is infected at time $t = 0$.

Theorem 3 *Given a time diffusion network $N_T = (V, \zeta)$, a set of friend nodes $F \subseteq V$, a set of malicious nodes $M \subseteq V$ and a time window t, the risk function $\rho_{N_T}(F', M, t)$ is monotonically nondecreasing and submodular in $F' \subseteq F$.*

Proof Given a sample $\{\delta_{ij}\}_{(i,j)\in N_T}$, we define $r_{\{\delta_{ij}\}}(F', M, t)$ as *the number* of nodes in M that can be reached from the nodes in F' at time less than or equal to t under $\{\delta_{ij}\}$; and $R_{\{\delta_{ij}\}}(u, M, t)$ as *the set* of nodes in M that can be reached from node u at time less than or equal to t under $\{\delta_{ij}\}$.

(i) $r_{\{\delta_{ij}\}}(F', M, t)$ is monotonically nondecreasing in F', for any sample $\{\delta_{ij}\}$. Indeed, $r_{\{\delta_{ij}\}}(F', M, t) = |\cup_{f\in F'} R_{\{\delta_{ij}\}}(f, M, t)|$ and so, for any $u \in V \setminus (F' \cup M)$, it holds that $r_{\{\delta_{ij}\}}(F', M, t) \leq r_{\{\delta_{ij}\}}(F' \cup \{u\}, M, t)$.

(ii) $r_{\{\delta_{ij}\}}(F', M, t)$ is submodular in F', for any sample $\{\delta_{ij}\}$. Let $R_{\{\delta_{ij}\}}(u|F', M, t)$ denote the set of nodes in M that can be reached from a friend node u in a time shorter than t, but that cannot be reached from any node in the set of friend nodes F' under $\{\delta_{ij}\}$. For any $F' \subseteq F'' \subseteq F$, it holds that $|R_{\{\delta_{ij}\}}(u|F', M, t)| \geq |R_{\{\delta_{ij}\}}(u|F'', M, t)|$. Consider now two sets of nodes $F' \subseteq F''(\subset F)$ and a node $u \in F \setminus F''$; then:

$$\begin{aligned} r_{\{\delta_{ij}\}}(F' \cup \{u\}, M, t) - r_{\{\delta_{ij}\}}(F', M, t) &= |R_{\{\delta_{ij}\}}(u|F', M, t)| \\ &\geq |R_{\{\delta_{ij}\}}(u|F'', M, t)| \\ &= r_{\{\delta_{ij}\}}(F'' \cup \{u\}, M, t) - r_{\{\delta_{ij}\}}(F'', M, t) \end{aligned}$$

If we average over the probability space of all possible transmission times, what we obtain (i.e., $\mathbb{E}[r_{\{\delta_{ij}\}}(F', M, t)]$, where $\mathbb{E}$ denotes the expectation) is also monotonically nondecreasing and submodular. If we denote with $\mathbb{I}$ the indicator function, we can conclude because

$$\begin{aligned} \mathbb{E}[r_{\{\delta_{ij}\}}(F', M, t)] &= \mathbb{E}[\sum_{m\in M} \mathbb{I}\{t_m(\{\delta_{ij}\}_{(i,j)\in N_T} | F') \leq t\}] \\ &= \sum_{m\in M} \mathbb{E}[\mathbb{I}\{t_m(\{\delta_{ij}\}_{(i,j)\in N_T} | F') \leq t\}] \\ &= \sum_{m\in M} Pr[t_m(\{\delta_{ij}\}_{(i,j)\in N_T} | F') \leq t] \\ &= \rho_{N_T}(F', M, t) \end{aligned}$$

Given a time diffusion network, if the risk function has a nonzero curvature, then the results of [1] also hold for this model. Let $S_{ij}(\delta_{ij})$ be the *survival function*, expressing the probability of v_j not being infected by node v_i in less than δ_{ij} time units. Formally, $S_{ij}(\delta_{ij}) := 1 - \int_0^{\delta_{ij}} f_{ij}(\delta')d\delta'$.

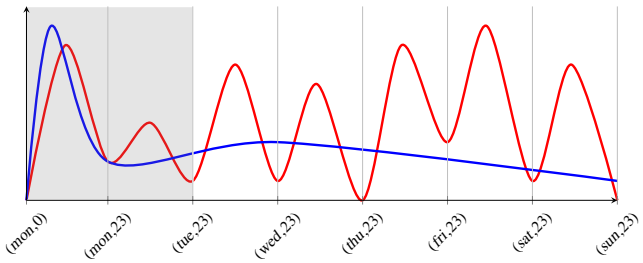


Fig. 8 Week distribution of ϕ_{Alice} in red and of ϕ_{Bob} in blue. Each interval represents a day, in grey is showed $(mon, 0, tue, 23)$.

Theorem 4 *Let $N_T = (V, \zeta)$ be a time diffusion network, for which $S_{ij}(\delta_{ij}) > 0$ until time t for all $v_i, v_j \in V$. Then $\kappa_{\rho(F,M,t)} > 0$.*

Proof The infection time of a given node in the network only depends on the transmission times drawn from the transmission functions. Thus, given a sample $\{\delta_{ij}\}_{(i,j)\in N_T}$, we first remove all $v_i \in F$ s.t. $\rho_{N_T}(\{v_i\}, M, t) = 0$, since they can be safely infected at time $t = 0$. Now pick an arbitrary $v \in F$, thus there exists a dipath P from v to some $v_m \in M$. Since by hypothesis the survival function is nonzero until time t for all pairs of nodes on the path, then $\prod_{(i,j)\in P} S_{ij}(\delta_{ij}) > 0$. This fact, entails that the likelihood of infection of every node on this path is decreased if this path is removed. Moreover, this implies $\rho_{N_T}(F, M, t) - \rho_{N_T}(F \setminus \{v\}, M, t) > 0$. Thus, by definition of curvature, we obtain $\rho_{N_T}(v|F \setminus \{v\}, M, t) > 0$ and therefore $\kappa_{\rho_{N_T}(F,M,t)} > 0$.

3.3 Absolute-Time Diffusion Networks

Let us consider a general diffusion network $N = (V, \gamma)$. We assume that each user in V is associated with a specification on his social-media behaviour, describing how often the user connects during a certain time period. Our reference period will be a generic week, but all results can be formulated by relying on other time periods. In this context, typical user behaviors are formulated as, e.g., "Alice usually logs-in on Saturday mornings" or "The time in which Bob stays connected during the evenings is the double than in the mornings".

We describe two different ways of formalize this specification resulting in two new models of diffusion networks: *concrete time diffusion networks* and *logical time diffusion networks*. We group them into the so-called *absolute time diffusion networks* because we assume the existence of a global indication of time passing.

3.3.1 Concrete Time Diffusion Networks

Assume that each user in V is associated with a continuous distribution describing the probability of the user log-in's during a general week.

Definition 20 (Concrete User Behaviour Describer, Concrete User Behaviour Variable Describer) A *concrete user behaviour describer* is a probability density

function denoted by *TFrame*. We define a *concrete user behaviour variable describer* as $\phi \sim TFrame$.

Definition 21 (Concrete Time Diffusion Network) A *concrete time diffusion network* is a tuple $N_{CT} = (V, \gamma, TF)$, where $V = \{v_i\}_{i=1\ldots n}$ is the set of nodes, $\gamma = (\gamma_{ij})_{i,j=1\ldots n}$ is the transmission matrix of the network (with $\gamma_{ij} \geq 0$, for all i, j) and $TF = (TFrame_i)_{i=1\ldots n}$ is the concrete user behaviour describer associated to each node.

For instance, assume that in our reference timeframe (the week) the distributions for Alice and Bob are the ones in Figure 8. This represents a scenario where "Alice's data traffic is higher in the middle hours of the days" and "Bob's data traffic is thickened on Monday mornings".

We can recover the structure of a general diffusion network from a concrete time diffusion network by considering the following approach. Let N_{CT} be a concrete time diffusion network. Look at user $u \in V$ which communicates with user $v \in V$. First of all, we need a way to formalize the time interval in which we want to track the evolution of the network. Let $days = \{mon, \ldots, sun\}$ be the set of days in a week s.t. $mon < tue < \cdots < sun$, and $hours = \{0, \ldots, 23\}$ be the set of hours in a day. Given (d_1, t_1) and (d_2, t_2), where $d_i \in days$ and $t_i \in hours$, we say that $(d_1, t_1) < (d_2, t_2)$ if $d_1 < d_2$ or $d_1 = d_2$ and $t_1 < t_2$.

Definition 22 (Concrete Interval) A *concrete interval* is a tuple (d_s, t_s, d_e, t_e) s.t. $(d_s, t_s) < (d_e, t_e)$. When $d_s = d_e$, we shall abbreviate (d_s, t_s, d_e, t_e) as $(d, [t_s, t_e])$.

Intuitively, d_s and d_e are respectively the starting day and the ending day of the interval, whilst t_s and t_e are respectively the starting time of day d_s and the ending time of day d_e. To make the presentation lighter, we divide the day in *phases*, whose granularity can be arbitrarily chosen (e.g., hour-by-hour, morning/afternoon/evening/night, AM/PM, ...). The idea is to encode the evolution of a network during a concrete interval through a Markov Chain, where:

1. *States* are $S = S_u \cup S_v \cup \{s_\emptyset\}$: we associate a state s_u and a state s_v for each phase of each day considered in the concrete interval. We consider also a state $s_\emptyset$ in which we go if u does not send messages or v does not receive them. We call S_u the *initials* states, S_v the *receiving* states and $S_f = S_v \cup \{s_\emptyset\}$ the *final* states of the MC.
2. Let $(d_1, ph_1) < \ldots < (d_h, ph_k)$ be the total order on the phases in which the concrete time interval is split; then, $Arcs = \left\{ \left(s_u^{d,ph}, s_v^{d',ph'} \right) : (d, ph) \leq (d', ph') \right\} \cup \left\{ \left(s_u^{(d,ph)}, s_u^{(d',ph')} \right) : (d, ph) \text{ and } (d', ph') \text{ are consecutive in the total order} \right\} \cup \left\{ \left(s_u^{d_h,ph_k}, s_\emptyset \right) \right\}$
3. $\mathscr{P} : Arcs \to [0,1]$ is the *transition probability* function s.t. $\sum_{(s,s') \in out(s)} \mathscr{P}(s,s') = 1, \forall s \in S_u$. By letting $s = s^{d,[t_1,t_2]}$ and $s' = s^{d',[t_1',t_2']}$, we have that

$$\mathscr{P}(s,s') = \begin{cases} P(\phi_u \in (d,[t_1,t_2]))P(\phi_v \in (d',[t_1',t_2'])) & \text{if } s \in S_u \text{ and } s' \in S_v \\ 1 - \sum\limits_{(s,x) \in out(s) \setminus \{(s,s')\}} \mathscr{P}(s,x), & \text{if } s, s' \in S_u, \text{ or } s \in S_u \text{ and } s' = s_\emptyset \end{cases}$$

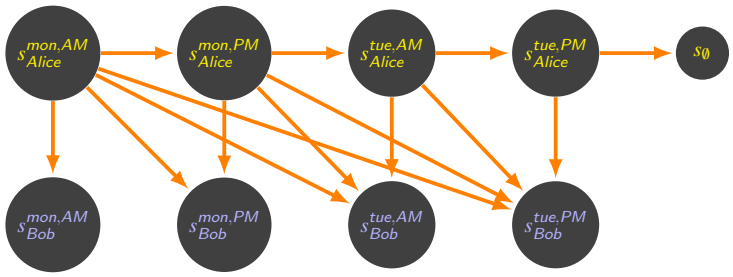


Fig. 9 Markov Chain for the example of Figure 8 in the concrete interval $(mon, 0, tue, 23)$.

4. $\iota : S_u \to [0,1]$ is the *initial distribution* such that $\iota(s) > 0$ is the probability that system evolution starts in s.

From now on, every time we write s_u or s_v, we actually mean $s_u^{d,[t_1,t_2]}$ or $s_v^{d',[t_1',t_2']}$. The idea is to label each arc (s,s') of the MC with $\mathscr{P}(s,s')$. If $s \in S_u$ and $s' \in S_v$, then $\mathscr{P}(s,s')$ gives us the probability that u sends a message to v in the concrete interval (d,t_1,d,t_2) and v receives the message in the interval (d',t_1',d',t_2'); otherwise, $\mathscr{P}(s,s')$ gives us the probability that u does not send a message to v in the interval (d,t_1,d,t_2).

Let $Path(s_u,s_v)$ be the set of all the paths starting in s_u and ending in s_v. The probability of reaching s_v starting from s_u is:

$$P(s_v|s_u) = \sum_{p \in Path(s_u,s_v)} \prod_{(s,s') \in p} \mathscr{P}(s,s').$$

Remark 3 Note that $0 \leq P(s_v|s_u) \leq 1$. Suppose $p_1 = (s_u,x_1) \rightsquigarrow (y_1,s_v)$ and $p_2 = (s_u,x_2) \rightsquigarrow (y_2,s_v)$ are any two paths in $Path(s_u,s_v)$. By construction we know $0 \leq \mathscr{P}(s_u,x_1) + \mathscr{P}(s_u,x_2) \leq 1$. Since for every edge (s,s') we have that $0 \leq \mathscr{P}(s,s') \leq 1$, then $\mathscr{P}(s_u,x_1) \geq \prod_{(s,s') \in p_1} \mathscr{P}(s,s')$ and $\mathscr{P}(s_u,x_2) \geq \prod_{(s,s') \in p_2} \mathscr{P}(s,s')$. This means that, if $0 \leq \mathscr{P}(s_u,x_1) + \mathscr{P}(s_u,x_2) \leq 1$, then also $0 \leq \prod_{(s,s') \in p_1} \mathscr{P}(s,s') + \prod_{(s,s') \in p_1} \mathscr{P}(s,s') \leq 1$.

Now, for each $s \in S_f$, we can calculate the probability of reaching such a state as follows:

$$P(s) = \sum_{s_u \in S_u} P(s_f|s_u)\iota(s_u).$$

Finally, we can recover a general diffusion network, by computing the probability of reaching any of the states in S_v starting from any of the states in S_u:

$$P(S_v|S_u) = \sum_{s_v \in S_v} P(s_v).$$

Remark 4 Clearly $0 \leq P(S_v|S_u) \leq 1$, since it represents the probability of reaching any of the state in $S_v \subset S_f$ and $P(S_f|S_u) = 1$.

Let us continue with the example of Figure 8 and assume that we split days in two phases: AM (from 00:00 to 11:59) and PM (from 12:00 to 23:59). Moreover, suppose we are tracking the trend of the net during the concrete interval $(mon,0,tue,23)$; then, we built the Markov Chain of Figure 9. We can now compute the probability of reaching each state in S_{Bob} as follows, where (d,AM) denotes the concrete interval $(d,0,d,11)$ and (d,PM) denotes $(d,12,d,23)$:

$$
\begin{aligned}
P(s_{Bob}^{mon,AM}) =& \iota(s_{Alice}^{mon,AM})(x_1y_1)\\
P(s_{Bob}^{mon,PM}) =& \iota(s_{Alice}^{mon,AM})[(x_1y_2)+(\bar{x_1}\cdot x_2y_2)]+\iota(s_{Alice}^{mon,PM})(x_2y_2)\\
P(s_{Bob}^{tue,AM}) =& \iota(s_{Alice}^{mon,AM})[(x_1y_3)+(\bar{x_1}\cdot x_2y_3)+(\bar{x_1}\cdot\bar{x_2}\cdot x_3y_3)]\\
&+\iota(s_{Alice}^{mon,PM})[(x_2y_3)+(\bar{x_2}\cdot x_3y_3)]+\iota(s_{Alice}^{tue,AM})(x_3y_3)\\
P(s_{Bob}^{tue,PM}) =& \iota(s_{Alice}^{mon,AM})[(x_1y_4)+(\bar{x_1}\cdot x_2y_4)+(\bar{x_1}\cdot\bar{x_2}\cdot x_3y_4)+(\bar{x_1}\cdot\bar{x_2}\cdot\bar{x_3}\cdot x_4y_4)]\\
&+\iota(s_{Alice}^{mon,PM})[(x_2y_4)+(\bar{x_2}\cdot x_3y_4)+(\bar{x_2}\cdot\bar{x_3}\cdot x_4y_4)]\\
&+\iota(s_{Alice}^{tue,AM})[(x_3y_4)+(\bar{x_3}\cdot x_4y_4)]+\iota(s_{Alice}^{tue,PM})(x_4y_4)
\end{aligned}
$$

where

$$
\begin{aligned}
x_1 =& P(\phi_{Alice}\in(mon,AM))=0.075 & x_2 =& P(\phi_{Alice}\in(mon,PM))=0.075\\
x_3 =& P(\phi_{Alice}\in(tue,AM))=0.045 & x_4 =& P(\phi_{Alice}\in(tue,PM))=0.045\\
y_1 =& P(\phi_{Bob}\in(mon,AM))=0.22 & y_2 =& P(\phi_{Bob}\in(mon,PM))=0.1\\
y_3 =& P(\phi_{Bob}\in(tue,AM))=0.05 & y_4 =& P(\phi_{Bob}\in(tue,PM))=0.07\\
\bar{x_1} =& (1-\sum_{i=1}^{4}x_1\cdot y_i)=0.967 & \bar{x_2} =& (1-\sum_{i=1}^{3}x_2\cdot y_i)=0,.97225\\
\bar{x_3} =& (1-\sum_{i=1}^{2}x_3\cdot y_i)=0.9856 & \bar{x_4} =& (1-x_4\cdot y_4)=0.99685
\end{aligned}
$$

Suppose that ι assigns equal probabilities to the states in S_{Alice}; then, the general diffusion network associated to $(mon,0,tue,23)$ is obtained by labelling the edge $(Alice,Bob)$ with

$$
\begin{aligned}
P(S_{Bob}|S_{Alice}) =& P(s_{Bob}^{mon,AM})+P(s_{Bob}^{mon,PM})+P(s_{Bob}^{tue,AM})+P(s_{Bob}^{tue,PM})\\
=& 0.0041+0.0111+0.0138+0.0363\\
=& 0.0653
\end{aligned}
$$

3.3.2 Logical Time Diffusion Networks

Information like that in Figure 8 is very detailed, but it is also quite difficult to obtain in its exact formulation. It is in practice much easier to give a more abstract description of the users' behaviors. For this reason, in this section we provide each user with a set of formulas describing his social behaviour. To this aim, we first introduce a new language called *users behaviour language*. Note that, creating an ad-hoc language for formulas in social networks is something also done in [9], where a general

logical framework called $\mathscr{L}_{\mathscr{S}\mathscr{N}}$ is introduced for reasoning about diffusion processes in social networks.

We need to set out the syntax and the semantics of $\mathscr{L}_{\mathscr{U}\mathscr{B}}$. Let us first define the alphabet. Like in the previous section, we shall consider a timeframe (the week, for example) and divide it in phases (e.g., in days, with every day split into AM and PM).

Definition 23 (Alphabet for $\mathscr{L}_{\mathscr{U}\mathscr{B}}$) Denote by $steps = \{s_1, \ldots, s_k\}$ the set containing the phases of the reference timeframe. The alphabet for $\mathscr{L}_{\mathscr{U}\mathscr{B}}$ is made by:

1. Connectives $\neg$, $\wedge$ and $\vee$;
2. Appartenance $\in$;
3. Parenthesis (and);
4. Constant symbols $s_1, \ldots, s_k$;
5. A step variable S ranging over *steps*.

Definition 24 (Syntax for Users Behaviour Language $\mathscr{L}_{\mathscr{U}\mathscr{B}}$) The syntax for *users behaviour language*, denoted $\mathscr{L}_{\mathscr{U}\mathscr{B}}$, is given by:

$$pre ::= S \in \Sigma \,|\, \neg pre \,|\, pre \wedge pre \,|\, pre \vee pre \,|\, (pre)$$
$$\phi ::= pre : p$$

where $p \in [0,1]$, $\Sigma \subseteq steps$ and $S \in \Sigma$ is the *atomic step formula*.

Notationally, we shall write $S = s$ in place of the atomic step formula $S \in \Sigma$ whenever $\Sigma = \{s\}$. We say that a step s satisfies a formula *pre*, and write $s \models pre$, if $pre[^s/_S] = true$. If $\phi = pre : p$, we define $steps(\phi) := \{s \in steps \ : \ s \models pre\}$.

Definition 25 (Logical User Behaviour Describer) A *logical user behaviour describer* Φ is a set of formulae ϕ s.t. :

1. $\forall s \in steps$, $\exists! \phi \in \Phi$ s.t. $s \in steps(\phi)$;
2. $\sum_{pre:p \in \Phi} p = 1$.

Definition 26 (Logical Time Diffusion Network) A *logical time diffusion network* is a tuple $N_{LT} = (V, \gamma, \Phi)$, where $V = \{v_i\}_{i=1\ldots n}$ is the set of nodes, $\gamma = (\gamma_{ij})_{i,j=1\ldots n}$ is the transmission matrix of the network (with $\gamma_{ij} \geq 0$, for all i, j) and $\Phi = (\Phi_i)_{i=1\ldots n}$ is the logical user behaviour describer associated to each node.

Like in the concrete time model seen in the previous section, we can turn a logical time diffusion network into a general diffusion network by computing a Markov Chain. As before, we can consider *steps* as pairs made up of a day and a time interval; then, the construction is like the one in the previous model, with the only difference in the definition of $\mathscr{P}$. Assume that $s = s^{(d,[t_1,t_2])}$ and $s' = s^{(d',[t_1',t_2'])}$; then,

$$\mathscr{P}(s,s') = \begin{cases} \frac{p}{|steps(\phi_u)|} \cdot \frac{q}{|steps(\phi_v)|} & \text{if } s \in S_u \text{ and } s' \in S_v \\ 1 - \sum_{(s,x) \in out(s) \setminus \{(s,s')\}} \mathscr{P}(s,x) & \text{if } s,s' \in S_u, \text{ or } s \in S_u \text{ and } s' = s_\emptyset \end{cases}$$

where $\phi_u = pre : p$ is the only formula of Φ_u such that $(d,[t_1,t_2]) \models pre$ and $\phi_v = pre' : q$ is the only formula of Φ_v such that $(d',[t'_1,t'_2]) \models pre'$.

Let Alice's and Bob's behaviours be described by "Alice's data traffic of the morning is twice as much as in the afternoon" and "Bob's data traffic is thickened on Monday morning". Suppose that $steps = days \times \{AM, PM\}$, i.e. *steps* contains pairs like (mon, AM), (mon, PM),..., $(sun, AM), (sun, PM)$. We define the following two sets of formulas:

$$\begin{aligned}\Phi_{Alice} = \{&S \in days \times \{AM\} : p_1,\\ &S \in days \times \{PM\} : p_2\}\\ \Phi_{Bob} = \{&S = (mon, AM) : q_1,\\ &S = (mon, PM) : q_2,\\ &\neg(S = (mon, AM) \vee S = (mon, PM)) : q_3\}.\end{aligned}$$

From the specifications above, we can deduce for Alice that $p_1 = 2p_2$; instead, we can describe Bob's behaviour, e.g., by the probabilities $q_1 = 2q_2$ and $q_2 = 2q_3$. Via Definition 25, we can solve these equations, thereby achieving: $p_1 = 0.66$, $p_2 = 0.34$, $q_1 = 0.57$, $q_2 = 0.28$ and $q_3 = 0.15$.

Hence, we have:

$$\begin{aligned}P(s_{Bob}^{mon,AM}) =& \iota(s_{Alice}^{mon,AM})(\frac{p_1}{7}q_1)\\ P(s_{Bob}^{mon,PM}) =& \iota(s_{Alice}^{mon,AM})[(\frac{p_1}{7}q_2) + (\bar{p_1} \cdot \frac{p_2}{7}q_2)] + \iota(s_{Alice}^{mon,PM})(\frac{p_2}{7}q_2)\\ P(s_{Bob}^{tue,AM}) =& \iota(s_{Alice}^{mon,AM})[(\frac{p_1}{7}\frac{q_3}{12}) + (\bar{p_1} \cdot \frac{p_2}{7}\frac{q_3}{12}) + (\bar{p_1} \cdot \bar{p_2} \cdot \frac{p_1}{7}\frac{q_3}{12})]\\ &+ \iota(s_{Alice}^{mon,PM})[(\frac{p_2}{7}\frac{q_3}{12}) + (\bar{p_2} \cdot \frac{p_1}{7}\frac{q_3}{12})] + \iota(s_{Alice}^{tue,AM})(\frac{p_1}{7}\frac{q_3}{12})\\ P(s_{Bob}^{tue,PM}) =& \iota(s_{Alice}^{mon,AM})[(\frac{p_1}{7}\frac{q_3}{12}) + (\bar{p_1} \cdot \frac{p_2}{7}\frac{q_3}{12}) + (\bar{p_1} \cdot \bar{p_2} \cdot \frac{p_1}{7}\frac{q_3}{12}) + (\bar{p_1} \cdot \bar{p_2} \cdot \bar{p_3} \cdot \frac{p_2}{7}\frac{q_3}{12})]\\ &+ \iota(s_{Alice}^{mon,PM})[(\frac{p_2}{7}\frac{q_3}{12}) + (\bar{p_2} \cdot \frac{p_1}{7}\frac{q_3}{12}) + (\bar{p_2} \cdot \bar{p_3} \cdot \frac{p_2}{7}\frac{q_3}{12})]\\ &+ \iota(s_{Alice}^{tue,AM})[(\frac{p_1}{7}\frac{q_3}{12}) + (\bar{p_3} \cdot \frac{p_2}{7}\frac{q_3}{12})] + \iota(s_{Alice}^{tue,PM})(\frac{p_2}{7}\frac{q_3}{12})\end{aligned}$$

where

$$\begin{aligned}\bar{p_1} =& (1 - \frac{p_1}{7}q_1 - \frac{p_1}{7}q_2 - 2\frac{p_1}{7}\frac{q_3}{12})\\ \bar{p_2} =& (1 - \frac{p_2}{7}q_2 - 2\frac{p_2}{7}\frac{q_3}{12})\\ \bar{p_3} =& (1 - 2\frac{p_1}{7}\frac{q_3}{12})\\ \bar{p_4} =& (1 - \frac{p_2}{7}\frac{q_3}{12})\end{aligned}$$

Suppose that ι assigns equal probability to the states in S_{Alice}; then, the general diffusion network associated to $(mon, 0, tue, 23)$ has the edge $(Alice, Bob)$ labeled with:

$$\begin{aligned} P(S_{Bob}|S_{Alice}) =& P(s_{Bob}^{mon,AM}) + P(s_{Bob}^{mon,PM}) + P(s_{Bob}^{tue,AM}) + P(s_{Bob}^{tue,PM}) \\ =& 0.0134 + 0.0406 + 0.0024 + 0.0066 \\ =& 0.063. \end{aligned}$$

4 The Gain-Risk Framework

In this section, we first introduce the gain function; this is somehow the counterpart of the risk function introduced in Section 2.2, since the gain counts the number of friend nodes (instead of the malicious ones, as done by the risk) that are expected to be infected. By exploiting this new function, we shall modify the policies MP and MU, by also making their definitions more symmetric. Finally, these new definitions will be used to simultaneously maximize the gain and minimize the risk, in the framework of multiobjective optimization.

4.1 Policy Enhancements

Let us consider a general diffusion network $N = (V, \zeta)$, with fixed and disjoint sets of friend nodes F and of malicious nodes M. We now give a new definition for when an initial infection $F' \subseteq F$ within a network satisfies a utility-restricted privacy policy or a privacy-restricted utility policy; to this aim, we first introduce the notion of *gain*.

Definition 27 (Gain) The *gain* $\pi(F', F, t)$ caused by $F' \subseteq F$ within time t is given by

$$\pi(F', F, t) = \sum_{f_i \in F} Pr[t_i \leq t | F']$$

Here, $Pr[t_i \leq t|F']$ is the likelihood that the infection time t_i of a friend node f_i is at most t, given that F' is infected at time $t = 0$.

Hence, the gain function is similar to the risk function but, instead of determining the expected number of infected nodes in M, it gives us the expected number of infected nodes in F. Clearly, since our definition of the gain function is similar to the definition of the risk function, we can state that:

1. the proof of submodularity for ρ in [1] can be easily adapted to show submodularity of π;
2. computing $\pi(F', M, t)$ is #P-hard;
3. the gain function can be approximated up to a constant factor, by following the algorithm in [18].

Moreover, we follow the approach in [1] for the risk function and assume to have an oracle that exactly computes the gain function for a given initial infection F'.

Definition 28 (Satisfaction of a Utility-restricted Privacy Policy) An initial infection F' *satisfies* a utility-restricted privacy policy $\Pi = (F,M,k,t)$ in a general diffusion network N if $F' \subseteq F$ and $\pi(F',M,t) \geq k$. A set F' *maximally satisfies* Π in N if there is no other set $F'' \subseteq F$ with $\pi(F'',F,t) \geq k$ and $\rho(F'',M,t) < \rho(F',M,t)$.

Definition 29 (Satisfaction of a Privacy-restricted Utility Policy) An initial infection F' *satisfies* an extended privacy-restricted utility policy $\Upsilon = (F,M,\tau,t)$ in a general diffusion network N if $F' \subseteq F$ and $\rho(F',M,t) \leq \tau$. A set F' *maximally satisfies* Υ in N if there is no other set $F'' \subseteq F$ with $\rho(F'',M,t) \leq \tau$ and $\pi(F'',F,t) > \pi(F',F,t)$.

For finding an initial infection meeting Definitions 28 and 29, we define the following problems.

Definition 30 (Extended Maximum k-Privacy - EMP) Given a utility-restricted privacy policy $\Pi = (F,M,k,t)$ and a general diffusion network N, the *extended maximum k-privacy problem* (EMP, for short) is given by

$$\begin{array}{ll} \underset{F' \subseteq F}{\text{minimize}} & \rho(F',M,t) \\ \text{subject to} & \pi(F',F,t) \geq k \end{array}$$

Definition 31 (Extended Maximum τ-Utility – EMU) Given a privacy-restricted utility policy $\Upsilon = (F,M,\tau,t)$ and a general diffusion network N, the *extended maximum τ-utility problem* (EMU, for short) is given by

$$\begin{array}{ll} \underset{F' \subseteq F}{\text{maximize}} & \pi(F',F,t) \\ \text{subject to} & \rho(F',M,t) \leq \tau \end{array}$$

Clearly, if F' is an optimal solution to the EMU problem with respect to Υ, then F' maximally satisfies Υ; similarly, if F' is an optimal solution to the EMP problem with respect to Π, then F' maximally satisfies Π. Unfortunately, EMP and EMU problems are NP-hard; this can be proved by reducing MP and MU to them.

Theorem 5 *EMU and EMP are NP-hard.*

Proof We just show the reduction of MU to EMU since the other one is symmetric. Let ϕ be an instance of the MU problem, we can construct an instance of the EMU problem ω by setting the time parameter of the gain function to $t = 0$. Hence, F' is the seed set of ϕ, respecting the risk constraint, iff F' is the maximum set of initially infected nodes always respecting the risk constraint. As the MU problem is NP-hard [1], also EMU is NP-hard.

However, like in the basic setting, both problems can be approximated, by slightly modifying Algorithms 1 and 2. For EMP, it suffices to replace line 1 in Algorithm 1 with

$1a: C \leftarrow \{X \subseteq F \ : \ \pi(X,F,t) \geq k\}$
$1b:$ **if** $C = \emptyset$ **then return** $\emptyset$ $\quad$ *//EMP cannot be satisfied.*

As discussed before, we can use the algorithms in [18] for the gain estimation: the randomized approximation algorithm for the influence estimation, when used as subroutine in the influence maximization algorithm, is guaranteed to find in polynomial time a set of X nodes with an influence of at least $(1-\frac{1}{e})OPT - 2X\varepsilon$, where ε is the accuracy parameter and OPT is the optimal value.

Similarly, Algorithm 2 can be easily adapted for handling EMU: it suffices to replace line 2 with

$$2:\ \tau' \leftarrow min_{F' \subseteq F}\, \rho(F', M, t) \ \text{ s.t. } \pi(F', F, t) \geq n.$$

4.2 The Gain-Risk Maximization Problem

We now exploit the the gain and the risk functions defined before to formulate a multiobjective maximization problem for a general diffusion network $N = (V, \zeta)$, with friends nodes F and malicious nodes M. First, we show that minimizing the risk ρ coincides with maximizing the quantity $|M| - \rho$; then, we formulate our multiobjective problem as a maximization of the two functions, π and $|M| - \rho$.

Lemma 1 *F^* is a solution for $\underset{F' \in 2^V}{\text{minimize}}\ \rho(F', M, t)$ if and only if it is a solution for $\underset{F' \in 2^V}{\text{maximize}}\ (|M| - \rho(F', M, t))$.*

Proof Let us denote with $(\star)$ the minimization problem and with $(\dagger)$ the maximization problem; we just show the *(Only if)* part, since the converse is symmetrical. By contradiction, suppose that F_1^* is a solution for $(\star)$ but not for $(\dagger)$, and let $F_2^* \neq F_1^*$ be a solution for $(\dagger)$. Then, we have $|M| - \rho(F_2^*, M, t) > |M| - \rho(F_1^*, M, t)$, that entails $\rho(F_2^*, M, t) < \rho(F_1^*, M, t)$; however, $\rho(F_1^*, M, t) < \rho(F_2^*, M, t)$: contradiction.

Definition 32 (Gain-Risk Maximization Problem) Given $\pi, \rho : 2^V \to \mathbb{R}$, the *gain-risk maximization problem* (GRMP) consists in

$$\underset{F' \in 2^V}{\text{maximize}} \quad \{\pi(F', F, t), |M| - \rho(F', M, t)\}. \tag{4}$$

GRMP clearly lies in the biobjective optimization problem class: we have two functions for which there may be not a single solution that maximizes the objective functions simultaneously. The idea is then to enumerate all the Pareto optimal solutions, since they represent the tradeoff between the two objective functions we are maximizing. However, such solutions could be exponentially many. Following the work in [12, 29], we avoid the complexity of enumerating all the possible Pareto optimal solutions by looking for a small family of representative solutions, by exploiting the notion of *regret ratio*.

F'	π	$3-\rho$	$p_{(1,0.5)}$	$p_{(0.5,1)}$	$p_{(0.5,0.5)}$
$\emptyset$	0	0	0	0	0
Bob	1	2	2	2.5	1.5
Cara	3	2	4	3.5	2.5
Bob & Cara	3	1	3.5	2.5	2

Table 1 The gain and risk values associated to each subset of $\{Bob, Cara\}$, and the corresponding values of the preference function computed by considering $\boldsymbol{a} \in \{(1,0.5),(0.5,1),(0.5,0.5)\}$.

	$p_{(1,0.5)}$	$p_{(0.5,1)}$	$p_{(0.5,0.5)}$
grr	0.5	1	0.5
$grrr$	0.125	0.285	0.2

Table 2 GR-regret and GR-regret ratio associated to $\{\{Bob\},\{Bob,Cara\}\}$ w.r.t. $p_{\boldsymbol{a}}$ under $2^{\{Bob,Cara\}}$, for $\boldsymbol{a} \in \{(1,0.5),(0.5,1),(0.5,0.5)\}$.

Regret Ratio in GRMP We now slightly adapt the definitions in [29] to our purposes. First, users' preference on utility/privacy is measured by a preference function.

Definition 33 (Preference Function, Vector of Preference) Given $\boldsymbol{a} \in \mathbb{R}^2_+$ (called *vector of preference*), the *preference function* $p_{\boldsymbol{a}}$ is a mapping $p_{\boldsymbol{a}} : 2^V \to \mathbb{R}_+$ defined as $p_{\boldsymbol{a}}(X) = a_1\pi(X,F,t) + a_2(|M| - \rho(X,M,t))$.

Intuitively speaking, if Alice is more worried about the utility aspect of the information spread, then the gain function will have a weight greater than the risk function (i.e., $a_1 > a_2$); on the other hand, if Bob is more focused on the privacy aspect, then the risk function will be more weighted (i.e., $a_2 > a_1$).

From now on, let us set $\mathscr{C} = 2^V$, let $\mathscr{S} \subseteq \mathscr{C}$ be a subfamily of $\mathscr{C}$, let p denote the preference function and $\boldsymbol{a}$ be its vector of preference.

Definition 34 (GR-Regret) The *GR-regret* of $\mathscr{S}$ with respect to $p_{\boldsymbol{a}}$ under $\mathscr{C}$ is $grr^{\mathscr{C}}_{p_{\boldsymbol{a}}}(\mathscr{S}) = \max\limits_{X\in\mathscr{C}} p_{\boldsymbol{a}}(X) - \max\limits_{X\in\mathscr{S}} p_{\boldsymbol{a}}(X)$.

Definition 35 (GR-Regret Ratio) The *GR-regret ratio* of $\mathscr{S}$ with respect to $p_{\boldsymbol{a}}$ under $\mathscr{C}$ is $grrr^{\mathscr{C}}_{p_{\boldsymbol{a}}}(\mathscr{S}) = \frac{grr^{\mathscr{C}}_{p_{\boldsymbol{a}}}(\mathscr{S})}{\max_{X\in\mathscr{C}} p_{\boldsymbol{a}}(X)} = 1 - \frac{\max_{X\in\mathscr{S}} p_{\boldsymbol{a}}(X)}{\max_{X\in\mathscr{C}} p_{\boldsymbol{a}}(X)}$.

Definition 36 (Maximum GR-Regret Ratio) The *maximum GR-regret ratio* of $\mathscr{S}$ under $\mathscr{C}$ is $mgrrr^{\mathscr{C}}(\mathscr{S}) = \max\limits_{\boldsymbol{a}\in\mathbb{R}^2_+} grrr^{\mathscr{C}}_{p_{\boldsymbol{a}}}(\mathscr{S})$.

Notice that the regret ratio is in the range $[0,1]$ and represents the normalized loss caused for the user in choosing a solution from $\mathscr{S}$ instead of $\mathscr{C}$. When the regret ratio is close to 0, the user is very happy, since choosing a solution from $\mathscr{S}$ gives a value that is close to the maximum; when the regret ratio is close to 1, the user is very unhappy, since choosing from $\mathscr{S}$ gives values that are far from the maximum.

For example, suppose that Alice absolutely wants to share her photo with Bob, Cara and as many friends as possible. She also wants the photo not to reach a certain group made by 3 users. Alice can initially send the photo to either Bob or Cara, to both of them, or to neither of them; so, $\mathscr{C} = \{\emptyset, \{Bob\}, \{Cara\}, \{Bob, Cara\}\}$ and suppose

$\mathscr{S} = \{\{Bob\}, \{Bob, Cara\}\}$. For simplicity, we assume that the only possible values of $\boldsymbol{a}$ are $(1, 0.5), (0.5, 1)$ or $(0.5, 0.5)$, and that at a certain time t the risk and the (complement of the) gain functions are depicted in the second and in the third column of Table 1; then, the last three columns of that table show the preference function calculated on each element of $\mathscr{C}$. Starting from this table, we compute in Table 2 the GR-regret and GR-regret ratio with respect to all possible values of $\boldsymbol{a}$. For instance, $grr^{\mathscr{C}}_{p_{(1,0.5)}}(\mathscr{S}) = 4 - 3.5 = 0.5$ and $grrr^{\mathscr{C}}_{p_{(1,0.5)}}(\mathscr{S}) = 0.5/4 = 0.125$. From Table 2 and under this restricted choices for $\boldsymbol{a}$, it is clear that $mgrrr^{\mathscr{C}}(\mathscr{S}) = 0.285$; this means that, if Alice chooses to initially share her photo with one of the sets in $\mathscr{S}$, the loss will be at most 0.285. For instance, if Alice worries more about privacy (so, she chooses the preference vector $\boldsymbol{a} = (0.5, 1)$), then her best choice in $\mathscr{C}$ is $\{Cara\}$, since $p_{(0.5,1)}(\{Cara\}) = 3.5$; looking in $\mathscr{S}$, she can select either $\{Bob, Cara\}$ or $\{Bob\}$ since $p_{(0.5,1)}(\{Bob, Cara\}) = p_{(0.5,1)}(\{Bob\}) = 2.5$. By contrast, if she worries more about utility (so, she chooses $\boldsymbol{a} = (1, 0.5)$), then her best choice in $\mathscr{C}$ is still $\{Cara\}$, whereas in $\mathscr{S}$ she has to select $\{Bob, Cara\}$, since $p_{(1,0.5)}(\{Bob, Cara\}) = 3.5 > 2 = p_{(1,0.5)}(\{Bob\})$.

The Algorithm. We solve GRMP by studying the following problem:

Definition 37 (Regret Ratio Minimization in GRMP) Given $\mathscr{C} = 2^V$ and $k \in \mathbb{N}$, find the $\mathscr{S} \subseteq \mathscr{C}$ with $|\mathscr{S}| \leq k$ that minimizes $mgrrr^{\mathscr{C}}(\mathscr{S})$.

Following [29], we associate $\mathscr{S} \subseteq \mathscr{C}$ with a polytope

$$P(\mathscr{S}) = \{\boldsymbol{x} \in \mathbb{R}^2_+ \mid \exists \boldsymbol{y} \in conv\{(\pi(X), |M| - \rho(X)) \mid X \in \mathscr{S}\} \text{ s.t. } \boldsymbol{x} \leq \boldsymbol{y}\}.$$

That is, we first consider the convex hull of the points in $\mathbb{R}^2$ which are images of the solutions in $\mathscr{S}$. Then, we consider the set of points dominated by at least one of the points in the convex hull. Our purpose is to approximate $Par(\mathscr{C})$, the curve of Pareto optima solutions, by respecting the standards of diversity and coverage given by the regret ratio minimization. This is done in Algorithm 3 that adapts Algorithm 2 from [29], which in turn is an adaptation of the *Chord* algorithm [12]. Intuitively, we shall iteratively compute $P(\mathscr{S})$ for approximating $Par(\mathscr{C})$ from the polytope's frontier faces.

We can better understand Algorithm 3 by looking at Figure 10. Here, the polytope is the area in red, at each iteration the points labelled with X are the ones in $\mathscr{S}$, the points in blue form the Pareto front. We suppose $k = 5$.

1. Figure 10(a): We start by finding the solutions X_1 and X_2, maximizing respectively π and $|M| - \rho$.
2. Figure 10(b): We find a nonnegative normal vector $\boldsymbol{a}$ of face X_1X_2 of $P(\mathscr{S})$.
3. Figure 10(c): We find the point X_3 by maximizing the preference function with respect to the vector $\boldsymbol{a}$; then, we recompute $P(\mathscr{S})$. Since $|\mathscr{S}| = 3$, we continue with the next iteration of the **while** loop.
4. Figure 10(d, e): We go on with the next iteration of the **while**. Now $P(\mathscr{S})$ has two faces; thus, we end up by adding X_4 and X_5 to $\mathscr{S}$. Since $|\mathscr{S}| = 5$, the algorithm terminates.

Algorithm 3 Gain-Risk Maximization Algorithm

Require: π and $|M|-\rho$ the objectives, $\mathscr{C}$ the ground set, $k \in \mathbb{N}$ the size of $\mathscr{S}$, *solve_MAX*$(f, \mathscr{X})$ the algorithm for solving $max_{X \in \mathscr{X}} f(X)$.
Ensure: *GRMP*$(\pi, |M|-\rho, \mathscr{C}, k)$
  $X_1 \leftarrow$ *solve_MAX*$(\pi, \mathscr{C})$
  $X_2 \leftarrow$ *solve_MAX*$(|M|-\rho, \mathscr{C})$
  $\mathscr{S} \leftarrow \{X_1, X_2\}$
  Compute $P(\mathscr{S})$
  while $|\mathscr{S}| < k$ **do**
    for each face F of $P(\mathscr{S})$ **do**
      Find a nonnegative normal vector $\boldsymbol{a}$ to F
      $X \leftarrow$ *solve_MAX*$(p_{\boldsymbol{a}}, \mathscr{C})$
      $\mathscr{S} \leftarrow \mathscr{S} \cup \{X\}$
      if $|\mathscr{S}| = k$ **then**
        return $\mathscr{S}$
    Compute $P(\mathscr{S})$
  return $\mathscr{S}$

5. Figure 10(f): The figure shows how the points in $\mathscr{S}$ approximate the Pareto front.

Notice that *solve_MAX* is any approximation algorithm for solving unconstrained maximization of submodular functions. An example is in [3], where it is showed a randomized algorithm achieving a tight $(1/2)$-approximation.

Analysis. The correctness of Algorithm 3 is proved by the adaptations of Lemma 2 and Theorem 10 in [29]; here we assume that *solve_MAX* is implemented via an approximated algorithm with an approximation factor α.

Lemma 2 *$mgrrr^{\mathscr{C}}(\mathscr{S}) \leq 1-\alpha$ if and only if $p_{\boldsymbol{a}}(X) \in \alpha^{-1}P(\mathscr{S})$, for all $X \in \mathscr{C}$ and $\boldsymbol{a} \in \mathbb{R}^2_+$.*

Theorem 6 *After Algorithm 3 investigates all the faces of P i times, $mgrrr^{\mathscr{C}}(\mathscr{S})$ is at most $1-\alpha+\sqrt{2}\cdot 2^{-i}$.*

Let us consider Figure 10(f), where we depicted in red $P(\mathscr{S})$ and in black $\alpha^{-1}P(\mathscr{S})$. As we can see, all the solutions in $\mathscr{S}$ are in a factor α from $P(\mathscr{S})$. Indeed, because Algorithm 3 is an adaptation of the *Chord* algorithm to submodular functions, we know that it approximates to a factor α the optimal solution. For example, if *solve_MAX* is an algorithm with approximation ratio of $1/2$ and $i=2$, the above theorem implies that the maximum gain-risk regret ratio is at most $1-1/2+\sqrt{2}\cdot 2^{-2} \approx 0.85$. That is, the loss caused in choosing from $\mathscr{S}$ instead of $\mathscr{C}$ is 0.85, i.e. the user will be quite unsatisfied. However, if we iterate the **while** loop 5 times, we almost reach the best possible upper bound on the maximum regret ratio that we can obtain with an (1/2)-approximation algorithm; to have better performances, and so to make the user more satisfied, we need a different algorithm for *solve_MAX*. This aspect is the real weakness of the whole approach: to our knowledge, there is no algorithm able of solving this problem in polynomial time when the cardinality of the ground set is exponential (like in our case, since $\mathscr{C} = 2^V$). Actually, the weakness derives also from the fact that the objective functions are submodular, whereas most of the literature considers multiobjective problems with *linear* functions. This aspect surely deserves more study in the future.

5 Conclusion

In this paper, we proposed some enhancements of the basic model in [1] for controlling utility and privacy in social networks. In particular, we added topics of conversation, time of the infection and user habits within the transmission likelihood. By means of a few small examples, we demonstrated the applicability of our enhanced frameworks on various situations that reflect aspects of real-life social networks. Then, we reduced all frameworks to the original one, so that the methods and results of [1] could be smoothly used in the settings we proposed. Furthermore, we modified the basic definitions of policy satisfaction, to make them closer to the intuitive meaning of such properties, thus introducing risk and gain functions. Finally, we tried to find a trade-off between such functions through multiobjctive optimization [10,26]. In particular, we used the notion of regret ratio to extract a family of representative solutions from the Pareto front. To validate our results, it would be interesting to setting up a few experiments on real-life data; this is left for future research.

References

1. Backes, M., Gomez-Rodriguez, M., Manoharan, P., Surma, B.: Reconciling Privacy and Utility in Continuous-Time Diffusion Networks. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF), pp. 292–304 (2017)
2. Brach, P., Epasto, A., Panconesi, A., Sankowski, P.: Spreading Rumours without the Network. In: COSN 2014 - Proceedings of the 2014 ACM Conference on Online Social Networks, pp. 107–118 (2014). DOI 10.1145/2660460.2660472
3. Buchbinder, N., Feldman, M., Naor, J., Schwartz, R.: A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization. In: 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science, pp. 649–658 (2012)
4. Budak, C., Agrawal, D., El Abbadi, A.: Limiting the Spread of Misinformation in Social Networks. In: Proceedings of the 20th International Conference on World Wide Web, pp. 665–674. ACM (2011)
5. Carnes, T., Nagarajan, C., Wild, S.M., van Zuylen, A.: Maximizing Influence in a Competitive Social Network: A Follower's Perspective. In: Proceedings of the Ninth International Conference on Electronic Commerce, pp. 351–360 (2007)
6. Chen, W., Wang, C., Wang, Y.: 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, pp. 1029–1038. ACM (2010)
7. Chen, W., Wang, Y., Yang, S.: Efficient Influence Maximization in Social Networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199–208. ACM (2009)
8. Chierichetti, F., Giakkoupis, G., Lattanzi, S., Panconesi, A.: Rumor Spreading and Conductance. J. ACM **65**(4), 17:1———17:21 (2018). DOI 10.1145/3173043. URL `http://doi.acm.org/10.1145/3173043`
9. Christoff, Z., Hansen, J.U.: A logic for diffusion in social networks. Journal of Applied Logic **13**(1), 48–77 (2015)
10. Cohon, J.L., Church, R.L., Sheer, D.P.: Generating multiobjective trade-offs: An algorithm for bicriterion problems. Water Resources Research **15**, 1001–1010 (1979)
11. Craft, D., Halabi, T., A Shih, H., R Bortfeld, T.: Approximating convex Pareto surfaces in multiobjective radiotherapy planning. Medical physics **33**, 3399–3407 (2006)
12. Daskalakis, C., Diakonikolas, I., Yannakakis, M.: How Good is the Chord Algorithm? SIAM Journal on Computing **45**(3), 811–858 (2016). URL `https://doi.org/10.1137/13093875X`
13. De Choudhury, M., Mason, W., M. Hofman, J., Watts, D.: Inferring relevant social networks from interpersonal communication. In: Proceedings of the 19th International Conference on World Wide Web, WWW '10, pp. 301–310 (2010)

14. Du, N., Song, L., Woo, H., Zha, H.: Uncover Topic-Sensitive Information Diffusion Networks. In: AISTATS, pp. 229–237 (2013)
15. Fujishige, S.: Submodular Functions and Optimization. Annals of Discrete Mathematics. Elsevier Science (2005)
16. Gomez-Rodriguez, M., Balduzzi, D., Schölkopf, B.: Uncovering the Temporal Dynamics of Diffusion Networks. In: ICML, pp. 561–568 (2011)
17. Gomez Rodriguez, M., Schölkopf, B.: Influence Maximization in Continuous Time Diffusion Networks. In: Proceedings of the 29th International Conference on Machine Learning, pp. 313–320. Omnipress (2012)
18. Gomez-Rodriguez, M., Song, L., Du, N., Zha, H., Schölkopf, B.: Influence Estimation and Maximization in Continuous-Time Diffusion Networks. ACM Trans. Inf. Syst. **34**(2), 9:1—-9:33 (2016)
19. Gorla, D., Granese, F., Palamidessi, C.: Enhanced Models for Privacy and Utility in Continuous-Time Diffusion Networks. In: R.M. Hierons, M. Mosbah (eds.) ICTAC 2019 - 16th International Colloquium on Theoretical Aspects of Computing, *Lecture Notes in Computer Science*, vol. 11884. Springer, Hammamet, Tunisia (2019). DOI 10.1007/978-3-030-32505-3_18. URL `https://hal.inria.fr/hal-02424329`
20. Iyer, R., Bilmes, J.: Submodular Optimization with Submodular Cover and Submodular Knapsack Constraints. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, pp. 2436–2444. Curran Associates Inc. (2013)
21. Kasprzak, R.: Diffusion in Networks. Journal of Telecommunications and Information Technology **nr 2**, 99–106 (2012)
22. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the Spread of Influence Through a Social Network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM (2003)
23. Lappas, T., Terzi, E., Gunopulos, D., Mannila, H.: Finding Effectors in Social Networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1059–1068. ACM (2010)
24. Nanongkai, D., Sarma, A.D., Lall, A., Lipton, R.J., Xu, J.: Regret-minimizing Representative Databases. Proc. VLDB Endow. **3**(1-2), 1114–1124 (2010)
25. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of Web sources. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, pp. 86–92 (2000)
26. Papadimitriou, C.H., Yannakakis, M.: Multiobjective Query Optimization. In: Proceedings of the Twentieth {ACM} {SIGACT-SIGMOD-SIGART} Symposium on Principles of Database Systems,, pp. 52–58. ACM (2001)
27. Richardson, M., Domingos, P.: Mining Knowledge-sharing Sites for Viral Marketing. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 61–70. ACM (2002)
28. Soma, T., Yoshida, Y.: Presentation of Regret Ratio Minimization in Multi-Objective Submodular Function Maximization. `https://www.opt.mist.i.u-tokyo.ac.jp/~tasuku/talks.html/` (2017)
29. Soma, T., Yoshida, Y.: Regret Ratio Minimization in Multi-Objective Submodular Function Maximization. In: AAAI, pp. 905—-911 (2017)
30. Tzoumas, V., Amanatidis, C., Markakis, E.: A Game-theoretic Analysis of a Competitive Diffusion Process over Social Networks. In: Proceedings of the 8th International Conference on Internet and Network Economics, WINE'12, pp. 1–14. Springer-Verlag, Berlin, Heidelberg (2012)
31. Watts, D., Dodds, P.: Influentials, Networks, and Public Opinion Formation. Journal of Consumer Research **34**, 441–458 (2007)
32. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998)
33. Yu, M., Gupta, V., Kolar, M.: An Influence-Receptivity Model for Topic Based Information Cascades. In: International Conference on Data Mining (ICDM), pp. 1141–1146. IEEE (2017)
34. Zeng, Y., Chen, X., Cong, G., Qin, S., Tang, J., Xiang, Y.: Maximizing Influence Under Influence Loss Constraint in Social Networks. Expert Syst. Appl. **55**(C), 255–267 (2016)
35. Zhou, D., Wenbao, H., Wang, Y.: Identifying Topic-sensitive Influential Spreaders in Social Networks. International Journal of Hybrid Information Technology **8**, 409–422 (2015)
36. Zhou, J., Zhang, Y., Cheng, J.: Preference-based mining of top-K influential nodes in social networks. Future Generation Computer Systems **31**, 40–47 (2014)

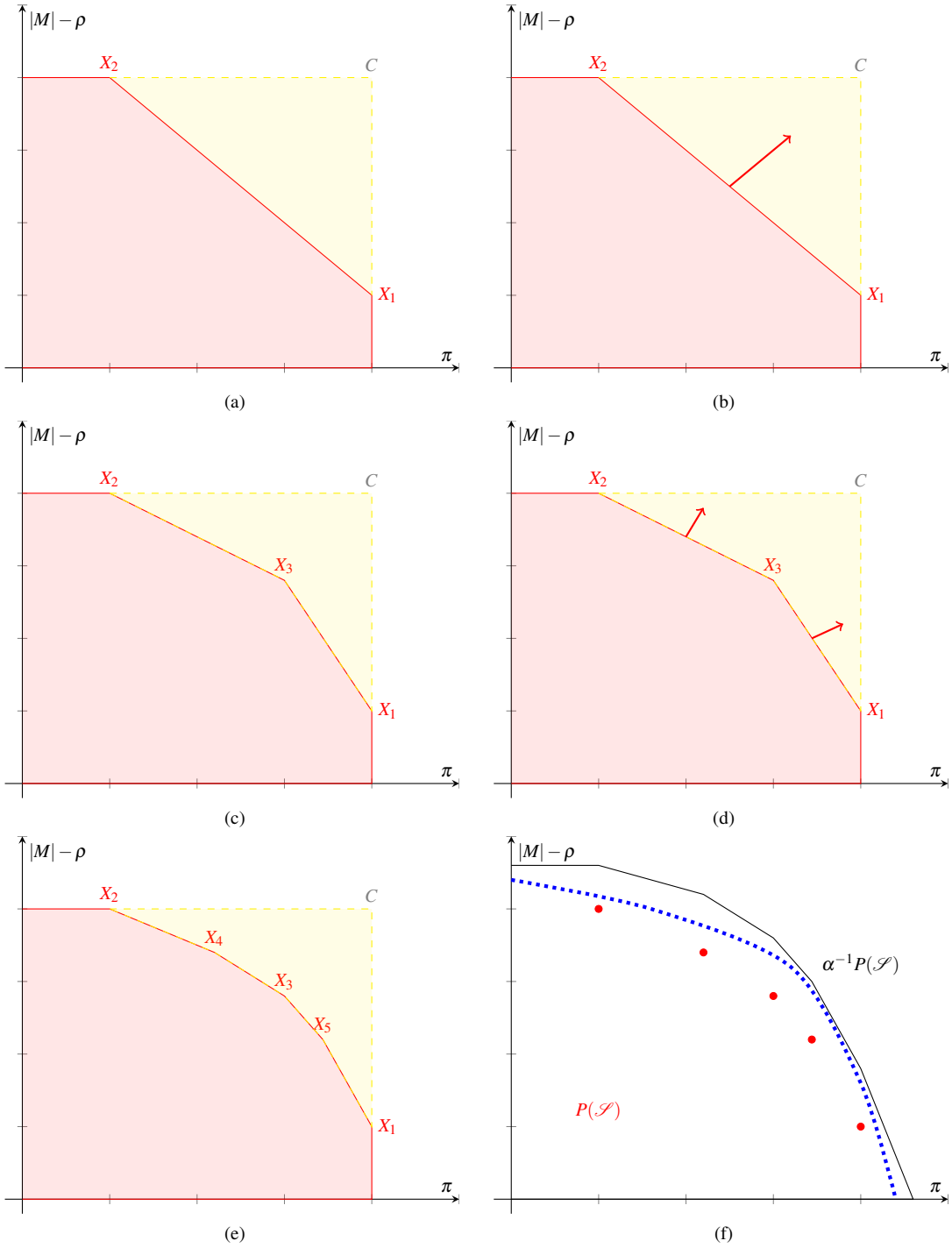


Fig. 10 Illustration of the Gain-Risk Maximization Algorithm (from [28]).