Competition, Collaboration and Optimization in Multiple Interacting Spreading Processes

Hanlin Sun\textsuperscript{1,*} and David Saad\textsuperscript{2,†}

\textsuperscript{1}University of Chinese Academy of Sciences, Beijing, China
\textsuperscript{2}Non-linearity and Complexity Research Group, Aston University, Birmingham B4 7ET, United Kingdom

Abstract

Competition and collaboration are at the heart of multi-agent probabilistic spreading processes. The battle on public opinion and competitive marketing campaigns are typical examples of the former, while the joint spread of multiple diseases such as HIV and tuberculosis demonstrates the latter. These spreads are influenced by the underlying network topology, the infection rates between network constituents, recovery rates and, equally importantly, the interactions between the spreading processes themselves. Here, we develop the theoretical framework for interacting spreading processes based on probabilistic dynamic message passing and optimize the infection or vaccination under budget constraints, within a finite time window; maximizing the desired spread in the presence of a rival competitive process, and limiting the spread through vaccination in the case of coupled infectious diseases. We demonstrate the efficacy of the framework and optimization method on both synthetic and realistic networks.

\* sunhanlin15@mails.ucas.ac.cn
\† d.saad@aston.ac.uk
I. INTRODUCTION

Political campaigns post the 2016 US presidential elections will never be the same.

Modern political and marketing warfare is taking place on social and information networks. Information supporting candidates, opinions, parties or products is passed between individuals occupying the network vertices in the form of a probabilistic process akin to epidemic spreading, where competing political parties, opinions or advertising agencies aim to get first to the largest possible audience. On the other hand, some spreading processes exploit previous exposure to one type of information (predetermined bias in the case of opinion setting or disease in the case of health-related epidemics) to facilitate the effective spread of another. Optimizing the use of limited resources for maximizing spread or for the containment of infectious diseases is clearly of great relevance but becomes more complex in the presence of multiple spreading processes that interact with one another.

Spreading processes have become increasingly more important in the fast moving modern world, where information is passed instantaneously on multilayered interwoven webs of contacts and where physical mobility is cheaper and easier than ever before. Consequently, pandemics, whether physical or virtual in the form of computer viruses, Internet rumors or marketing campaigns spread like wildfires. For instance, in 2017, a worldwide cyberattack by the WannaCry ransomware cryptoworm was estimated to have affected more then 200,000 computers across 150 countries, with total damages ranging from hundreds of millions to billions of dollars [1]. In 2018, a small-scale outbreak of African swine fever, caused by Asfivirus in China [2], has posed the risk of spreading globally.

The case of multiple interacting spreading processes, competitive or collaborative is particularly challenging due to the interaction between processes that affects the spreading process in non-trivial ways. In competitive processes the interaction between processes in inhibitory while in collaborative processes it is excitatory. An example for a competitive process can be the spreading of rumors. Recently, the “Anti-Vaxx movement” in the US, has attracted the attention of the public, for instance through Twitter messages, leading to the belief of a growing number of parents that vaccination is a violation of human-rights, that vaccines cause autism, brain damage and do not benefit the health and safety of society. The idea has been spreading rapidly through social media; as a result the measles virus, which was declared to be eliminated in 2000, is making a comeback [3]; consequently, the World Health Organization declared Vaccine hesitancy as one of the top 10 global threats [4] and social network platforms have been requested to block the spread of related information [5].

If both equally probable “valid information” and “unsubstantiated rumors” spread on the network simultaneously, individuals exposed to one tend to believe in its content and are less susceptible to the other. An example for a collaborative spreading process is the co-infection of HIV and tuberculosis, the latter being the main cause of death of AIDS patients. The risk of developing tuberculosis is estimated to be 16-27 times higher in people living with HIV than among uninfected individuals [6]. AIDS patients are more susceptible to tuberculosis, due to their weakened immune system, and tuberculosis can also activate the replication of HIV virus. Epidemiological studies have also shown that the co-infection also exists between HIV and malaria parasites [7], and Zika and Dengue viruses [8].

Two important aspects to be studied in all spreading processes, are forecasting the dynamics and the optimal use of resources to either maximize or minimize the spread. Forecasting is based on a probabilistic modeling and inference of the vertices’ state at any given time and optimization requires the development of optimization methods for the deployment of resources over time to meet the objectives. For instance, in a single epidemic spreading process, prediction can describe the infection probability at any time given initial conditions, while optimization could refer to the initial choice of best spreaders or to the best vaccination strategy to control and prevent an outbreak of a
There exists a large body of work on probabilistic modeling of spreading processes and their optimization. Most analyses rely on the probabilistic transition of system constituents from state to state depending on their original state and that of their neighbors. Most commonly used are the SIR (Susceptible, Infected, Recovered), SI (Susceptible, Infected) and SIS (Susceptible, Infected, Survived) models, which correspond to specific scenarios. They have been analyzed using a variety of methods [9–12], most relevant to the current work are methods based on belief propagation and the cavity method [13–15].

Optimal resource deployment in various spreading settings has also been investigated in many scenarios in the case of a single spreading process. One of the most commonly studied scenarios is that of identifying the most influential spreaders, on which the deployment of resource at time zero would maximize the spread at a given end time. Most of these studies rely on the network’s topological properties and selection strategies are based on high-degree nodes [16], neighbors of randomly selected vertices [17], betweenness centrality [18], random-walk [19], graph-partitioning [20], and k-shell decomposition [21]. These approaches mostly ignore important dynamical aspects that impact on performance [22, 23]. A related approach termed network dismantling [24–26] aims at identifying the nodes which, if removed, lead to the fragmentation of the giant component and prevent the global percolation. The optimal deployment of immunization has been addressed using a belief propagation algorithm [27], based on cavity method techniques developed previously for deterministic threshold models [13, 14].

Several scenarios that incorporate the dynamical properties of the spreading process, such as optimal seeding, which allocates the set of initial nodes that maximizes the spread asymptotically have been studied and analyzed [28–30] for the simple diffusion models of Independent Cascade (IC) and Linear Threshold, and have been shown to be NP-hard for both. A more challenging task is the study of scenarios with a finite time horizon as studied for the IC [31] and other spreading models [32].

Most relevant to the current study is the application of a recurrent optimization framework [33] to the Dynamic Message Passing (DMP)-based probabilistic modeling of spreading processes [15]. The framework also facilitate both open-loop resource allocation (a one-off preplanned assignment) and a closed-loop (iterative) real-time dynamical resource deployment under a limited remedial budget. We utilize a similar framework to investigate and optimize the dynamics of multi-agent spreading processes.

The analysis of multi-agent processes is much more involved due the interaction between processes and its impact on the spreading dynamics. Existing numerical studies [34] have revealed new transitions and a more aggressive spreading mode, which points to a percolation transition in cooperative contagion and highlights the increased risk of an unpredictable and violate outbreak in such scenarios. The most relevant studies to the current work focus on the analysis of multi-agent spreading processes in a competitive scenario on a specific network, using continuous equations similar to those of dynamic message-passing [35]; and on a two-stage infection process, which is a specific case for multi-agent spreading processes [36].

To the best of our knowledge, no successful analysis and optimization have been offered to address the general case of multi-process modeling and optimization. Moreover, as most optimization algorithms for single-agent processes are based on using the network topology, a static network property, they cannot fully capture the intricate dynamics of multi-agent processes and are therefore less effective for optimization tasks.

In this paper, we generate a standard framework for probabilistic forecasting and optimization in both competitive and collaborative scenarios, which is based on the DMP algorithm [15, 33]. We demonstrate that the framework
provides an accurate description of both competitive and collaborative scenarios on both toy and large-scale problems; it is exact on tree-like networks and provides a good approximation on networks with loops. We then develop an optimization algorithm for maximizing the spread within a given time window against a competing spread as well as the containment of spreads in a collaborative-spreadings scenario through an optimized vaccination strategy that curbs one of the spreading processes. We demonstrate the efficacy of the suggested algorithm, offering good results with a tractable computational cost.

The paper is organized as follows: In Section II we will derive the DMP-based model for multiple interacting spreading processes. In Section III, we validate the efficacy of the probabilistic model by comparing the results obtained with those of Monte Carlo simulations on synthetic and real networks. The optimization algorithm is introduced in Sec. IV for both competitive and cooperative scenarios and is tested on synthetic networks in Sec. V A. In Section V B we apply the optimization algorithm to real world networks to demonstrate its usefulness to more realistic scenarios; these include both competitive and collaborative cases, and the optimal deployment of vaccines to contain an epidemic. A summary and outlook are provided in Sec. VI.

II. MODEL

The models investigated in this work are based on the SI and SIR processes, but where a number of processes are active in parallel and interact with one another. The implication is that the state of a network vertex determines its susceptibility to be infected by the spreading processes. For instance, in mutually exclusive competitive processes, once a vertex has been infected by one process it cannot be infected by any other process and retains its state, which is also termed “cross-immunity” [36]. In a collaborative spreading scenario, being infected by one process increases the susceptibility of being infected by other processes according to some predefined conditional probability. Although the framework for the various scenarios is similar it does include some important differences and will therefore be developed separately.

A. Mutually Exclusive Competitive Processes

Numerous multi-process spreading models do feature in the real world. For instance, convincing an individual to buy a specific product or adopt a certain opinion suppresses their probability of considering a competitive product/opinion. For simplicity, we choose a representative two-process competitive SI model, described as follows: A graph $G = (V, E)$ comprises the set of vertices $V$ and edges $E$ such that each node (vertex) $i$ can be found in one of three states at any time $t$: susceptible ($\sigma_i(t) = S$), infected by disease $A$, ($\sigma_i(t) = A$), or infected by disease $B$, ($\sigma_i(t) = B$). The infection rules are as follows:

\[
A(i) \xrightarrow{} B(i), \quad B(i) \nrightarrow A(i)
\]

\[
S(i) + A(j) \xrightarrow{p} A(i) + A(j)
\]

\[
S(i) + B(j) \xrightarrow{q} B(i) + B(j)
\]

(1)

In other words, two infection processes $A$ and $B$ are mutually exclusive and any node can be infected by a neighboring nodes, assuming one of the two states, but then cannot change its state. Since the infection is edge-based and a two-vertex interaction, the processes in (1) seem deceptively as two completely independent parallel processes; however,
they clearly interact through the graph topology and the exclusivity of the adopted states. Indeed, we assume that at any given time step, the infection probabilities of process $A$ (denoted $p$) or process $B$ (denoted $q$) are independent, but the probability of being infected by both $A$ and $B$ simultaneously is forced to be 0. For closing the equations we need to define what happens in the case where both processes infect the vertex, resulting in an invalid state. The two main options considered are: a) Resampling - Since infection by both processes is invalid it is set to zero and once have computed the unnormalized probabilities $\hat{P}(\sigma^i_A(t+1)|\sigma^i_S(t))$, $\hat{P}(\sigma^i_B(t+1)|\sigma^i_S(t))$, $\hat{P}(\sigma^i_{AB}(t+1)|\sigma^i_S(t))$, we carry out a renormalization procedure

$$P(\sigma^i_{A/B/S}(t+1)|\sigma^i_S(t)) = \frac{\hat{P}(\sigma^i_{A/B/S}(t+1)|\sigma^i_S(t))}{\hat{P}(\sigma^i_S(t+1)|\sigma^i_S(t)) + \hat{P}(\sigma^i_A(t+1)|\sigma^i_S(t)) + \hat{P}(\sigma^i_B(t+1)|\sigma^i_S(t))}$$

(2)

to make sure that

$$P(\sigma^i_A(t+1)|\sigma^i_S(t)) + P(\sigma^i_B(t+1)|\sigma^i_S(t)) + P(\sigma^i_{AB}(t+1)|\sigma^i_S(t)) = 1,$$

(3)

enforcing $P(\sigma^i_{AB}(t+1)|\sigma^i_S(t)) = 0$.

This corresponds to the case in simulating the process whereby a sampled case of infection by both processes $A$ and $B$ is ignored and resampling is carried out. b) Probability redistribution - Since the state $AB$ is invalid, one may assume that one of the processes dominates, probabilistically, and distribute the probability of invalid cases $\hat{P}(\sigma^i_{AB}(t+1)|\sigma^i_S(t))$ according to the probabilities $\hat{P}(\sigma^i_A(t+1)|\sigma^i_S(t))$ and $\hat{P}(\sigma^i_B(t+1)|\sigma^i_S(t))$. Carrying out the renormalization procedure

$$P(\sigma^i_A(t+1)|\sigma^i_S(t)) = \hat{P}(\sigma^i_A(t+1)|\sigma^i_S(t)) \left[ 1 + \frac{\hat{P}(\sigma^i_{AB}(t+1)|\sigma^i_S(t))}{\hat{P}(\sigma^i_A(t+1)|\sigma^i_S(t)) + \hat{P}(\sigma^i_B(t+1)|\sigma^i_S(t))} \right],$$

(4)

and similarly for $P(\sigma^i_B(t+1)|\sigma^i_S(t))$ by exchanging $A \leftrightarrow B$, ensures that the probabilities sum to 1.

While both options can be incorporated in the analysis we have chosen to use option a), due to its simplicity and straightforward implementation is simulations.

We will use the framework of DMP to transform equations (1) and (3) to a closed set of message-passing equations with respect to a new set of variables, following the derivation of [15, 33]. The method is based on the cavity method of statistical physics whereby probabilistic messages are passed along the graph edges and the marginal probabilities are based on the consistent set of messages. The obtained marginals are exact on tree graphs and provide a good approximation on locally tree-like sparse graphs with loops. We provide an intuitive derivation of the DMP equations for the generalized SI model of Eqs. (1).

On a given instance of a network, these equations allow one to compute the marginal probability distributions $P^i_S(t)$, where $\sigma \in \{S, I_A, I_B\}$ denotes the state of the node, based on probabilistic messages from its neighbors. For simplicity, we make a brief introduction of the idea of dynamic message-passing. The detailed equations for the competitive scenario appear in Appendix A.

The aim of the DMP equations is to calculate a good approximation to the posterior for each node through message-passing in time. The first key equation for the posterior $\hat{P}_S^i(t)$ reads:

$$\hat{P}_S^i(t) = P_S^i(t-1) \prod_{k \in h} \frac{\theta^{k \rightarrow i}_A(t)\theta^{k \rightarrow i}_B(t)}{\theta^{k \rightarrow i}_A(t-1)\theta^{k \rightarrow i}_B(t-1)} \tag{5}$$
The equation states that the probability of node $i$ to be susceptible at time $t$ equals the probability (prior to renormalization) it was in state $S$ at time $t-1$, $\hat{P}_S^{i}(t-1)$ and remained so until time $t$, and was not infected at this time by a neighbor, neither by process $A$ nor $B$. The dynamic message $\theta^{k\rightarrow i}_A(t)$ has a meaning of the probability that node $k$ did not pass an activation message (infection) $A$ to node $i$ until time $t$ (similarly for $B$). Meanwhile, since $i$ was in state $S$ at time $t-1$, which has implied $i$ was not infected by a neighbor, neither by process $A$ nor $B$ at time $t-1$, this condition needs to be considered when calculating $\hat{P}_S^{i}(t)$, hence the added denominator. Equation (5) is exact only on tree graphs as it relies on the probabilistic independence of all $k \in \partial i \setminus j$ and factorization; nevertheless it provides a good approximation on general sparse graphs even with small loops [15].

Similar equations have been derived for $\hat{P}_A^{i}(t)$ and $\hat{P}_B^{i}(t)$ and are presented in Appendix A, Eq.(A9). The renormalized posterior values are calculated on the basis of Eq.(2). The quantities $\theta^{k\rightarrow i}_A(t)$ and $\theta^{k\rightarrow i}_B(t)$ are updated in this case as follows:

$$\theta^{k\rightarrow i}_A(t) = \theta^{k\rightarrow i}_A(t-1) - p\phi^{k\rightarrow i}_A(t-1)$$
$$\theta^{k\rightarrow i}_B(t) = \theta^{k\rightarrow i}_B(t-1) - q\phi^{k\rightarrow i}_B(t-1),$$

It corresponds to the fact that $\theta^{k\rightarrow i}_A(t)$ decreases if an activation signal is passed along the directed link $(ki)$; the corresponding probability equals the product of $p$ and the dynamic variable $\phi^{k\rightarrow i}_A(t-1)$, which has a meaning of the probability that node $k$ is in state $A$ at time $t-1$, but has not infected node $i$ until time $t-1$. To simplify the explanation we introduce the dynamic messages $P_S^{i\rightarrow j}(t), P_A^{i\rightarrow j}(t)$ and $P_B^{i\rightarrow j}(t)$, which denote the probabilities that node $k$ is found at time $t$ in the states $S$, $A$ or $B$, respectively, conditioned on node $i$ remaining in state $S$. Alternatively, these variables can be thought of as the probabilities of $k$ being susceptible, infected by processes $A$ or $B$ on a cavity graph, in which node $i$ has been removed.

The messages on the cavity graph $\hat{P}_{S/A/B}^{i\rightarrow j}(t)$ can be derived in a similar way to that of the posterior (5), for instance for $S$:

$$\hat{P}_{S}^{i\rightarrow j}(t) = P_S^{i\rightarrow j}(t-1) \prod_{k \in \partial i \setminus j} \frac{\theta^{k\rightarrow i}_A(t)\theta^{k\rightarrow i}_B(t)}{\theta^{k\rightarrow i}_A(t-1)\theta^{k\rightarrow i}_B(t-1)}$$

Similar equations have been derived for $\hat{P}_A^{i\rightarrow j}(t)$ and $\hat{P}_B^{i\rightarrow j}(t)$ and are presented in Appendix A, Eq.(A7). The renormalized message values are calculated in a similar way to Eq.(2).

To close the set of coupled equations we write the dynamical equation for $\phi^{k\rightarrow i}_{A/B}(t)$:

$$\phi^{k\rightarrow i}_A(t) = (1-p)\phi^{k\rightarrow i}_A(t-1) + \Delta P^{k\rightarrow i}_A(t-1)$$

where $\Delta P^{k\rightarrow i}_A(t-1) \equiv P^{k\rightarrow i}_A(t) - P^{k\rightarrow i}_A(t-1)$. The physical meaning of equation (8) is as follows: $\phi^{k\rightarrow i}_A(t)$ decreases if the activation signal is actually transmitted (first term) and increases if node $k$ transitions to state $A$ at the current time step. Similar equations should be written for process $B$. Equations (7)-(8) can be iterated in time starting from the given initial conditions $\{P_S^{i}(0), P_A^{i}(0), P_B^{i}(0)\}_{i \in V}$, with

$$\theta^{i\rightarrow j}_{A/B}(0) = 1, \quad \phi^{i\rightarrow j}_{A/B}(0) = \delta_{i,A/B} = P_{A/B}^{i}(0).$$

The marginals $P_S^{i}(t), P_A^{i}(t)$ and $P_B^{i}(t)$ are obtained using (7) and the corresponding equations for the spreading processes $A$ and $B$, while employing the normalization condition (3). The computational complexity of the DMP equations for solving the dynamics up to time $T$ is given by $O(|E|T)$, where $|E|$ is the number of edges in the graph, which makes them scalable for sparse networks with millions of nodes. Details of the derivation and equations obtained appear in Appendix A.
B. Collaborative Process

In the collaborative scenario we consider here there are two processes and when a node is infected by one of them it can be more (or less) susceptible to be infected by another. A node can be infected by both processes and the influence of the different processes is not necessarily symmetric.

The interaction in collaborative processes can be denoted by listing the possible transitions:

\[ S(i) + A(j) \xrightarrow{p_A} A(i) + A(j) \]
\[ S(i) + B(j) \xrightarrow{p_B} B(i) + B(j) \]
\[ A(i) + B(j) \xrightarrow{p_{AB}} A(i) + AB(j) \]
\[ A(i) + B(j) \xrightarrow{p_{BA}} AB(i) + B(j) \]

In this scenario, state \( AB \) can be regarded as a combination of state \( A \) and state \( B \). In principle, there is no difference between collaborative processes and competitive processes, since the latter can be regarded as a special case of the former. It should be noted that other dependencies can be easily accommodated within a similar framework but should be clearly expressed and joint/conditional probabilities. However, here we focus on the collaborative case where \( p_{BA} \) and \( p_{AB} \) are larger than \( p_A \) and \( p_B \), i.e., if a node is infected by one process and it is more vulnerable to another, and vice versa.

For completeness we have to determine how to deal with the process \( S(i) \rightarrow AB(i) \). In mutually exclusive processes there is clearly no such transition and

\[ S(i) \not\rightarrow AB(i) \]

while in collaborative processes, we allow for an infection of the form

\[ S(i) + AB(j) \xrightarrow{p_A \times p_B} AB(i) + AB(j) \],

where the infection processes for A and B are considered as independent. This leads to significant differences in details of the dynamic message passing equations provided in Appendix B.

III. VALIDATION OF MODELS

To validate the DMP equations obtained for competitive/collaborative processes we test the accuracy of the inferred node values against numerical results obtained via Monte Carlo simulations. Testing will be carried out on both synthetically generated networks and real instances.

A. Inference in Competitive Spreading

Validation is carried out on two synthetic networks generated by package NetworkX, a tree network and a network with loops; and two real world benchmark networks, the Polbooks and Football [37] examples. With the given initial condition, we implement both Monte Carlo simulation and DMP method on the two models.
Exhaustive numerical experiments reveal that the improvement in performance due to the use of DMP reduces when the value of the infection parameters is very high. Therefore, we choose reasonable infection parameters in our examples and do not examine cases with extreme values. The location of seeds initializing the processes and target nodes to be observed are chosen randomly.

In Fig. 1 we show two synthetically generated networks (a) a toy tree network of 10 nodes and (b) a network of 10 nodes with loops. The other networks presented in (c) and (d), the Football [37] and Polbooks [38] networks, with 115 and 105 nodes respectively, are standard benchmark networks.

To validate the efficacy of DMP in modeling competitive scenarios we compare results obtained from running equations (A4)-(A9) against results obtained using Monte Carlo simulations. Simulations are carried out 10 times for gathering statistics, each round includes $10^3$ samples per node (about $10^5$ samplings in total, depending on the network size). The parameters used in the toy model tests are $p = 0.3$ and $q = 0.7$, and we observed the marginal posterior probabilities $P_i^A(t)$ in both Figs. 2(a) and (c), and $P_i^S(t)$ in Figs. 2(b) and (d) for the tree and loopy graphs, respectively. The seeds initializing the processes are placed in nodes 7 for process $A$ and in nodes 2 for process $B$. The choice of this particular node has been arbitrary and has no significance. The results obtained show excellent agreement between theory and simulations. We also tested the accuracy of the method on the benchmark Polbooks network as shown in Fig. 2(e) and (f) for the parameters $p = 0.2$, $q = 0.2$. In this case, process $A$ starts from nodes 1 and 2, process $B$ from nodes 4 and 37 (again, both are arbitrary choices). The observed probabilities $P_i^A(t)$ in Fig. 2(e) and $P_i^S(t)$ in Fig. 2(e) show good agreement between theory and simulations.
FIG. 2. Comparison of DMP-based results and Monte Carlo simulations for competitive processes. For each of the graphs $10^5$ samples have been used and both mean values and error-bars are shown, unless they are smaller than the symbol size. (a) For a competitive process on a tree-like network using the parameters $p = 0.3$, $q = 0.7$ and observing $P_{A}^{i=3}(t)$. The seeds initializing the processes are placed in nodes 7 for process $A$ and in nodes 2 for process $B$. (b) The same as in (a) but observing $P_{S}^{i=3}(t)$. (c) Comparing results from DMP method and Monte Carlo simulation for the network with loops using the same as (a) and observing $P_{A}^{i=3}(t)$. (d) The same as (c) but observing $P_{S}^{i=3}(t)$. For the Polbooks experiment we observe (e) $P_{A}^{i=0}(t)$ and (f) $P_{S}^{i=0}(t)$; the parameters used are $p = 0.2$, $q = 0.2$ and the seeds initializing the processes are placed in nodes 1 and 2 for process $A$ and in nodes 4 and 37 for process $B$. 
B. Inference in Collaborative Spreading

Similar experiments were run for a collaborative process on the toy tree network, graphs with loops and the benchmark Football network. The results show good agreement with between theory and simulations, and are presented in detail in Appendix C.

IV. OPTIMIZATION

Competition and collaboration of spreading processes on graphs can be optimized or contained through a judicious use of resource. We will demonstrate how managing a spreading process against an adversarial competing process can be optimized within a given time frame; and how joint collaborative processes can be affected through best use of resource. The latter can take the form of spreading maximization while making use of the process interdependencies, or of containment through vaccination to impact on the spread of both processes.

We illustrate a general procedure for optimization in this section. Details for specific optimization problems we address here are given in Appendix D for competitive processes and in Appendix F for collaborative processes.

The core approach for optimization is based on a discretized variational method, whereby a functional over a time window (Lagrangian) is optimized through changes in control parameters throughout the time interval. The dynamics, resource restrictions, initial conditions and other restrictions on the parameters used are enforced through the use of Lagrange multipliers. A similar method is used in optimal control.

We denote the components of the Lagrangian function used in a similar way to [33]:

$$L = \sum \left( \mathcal{O} + B + \mathcal{I} + \mathcal{P} + \mathcal{D} \right).$$  \hspace{1cm} (16)

where in the $L$ is the Lagrangian function, $\mathcal{O}$ is the objective to be optimized, $B$ is the budget or constraints on the resource used, $\mathcal{I}$ represents the component that forces initial conditions, $\mathcal{P}$ are restrictions on the probabilities used and $\mathcal{D}$ represent the dynamics as derived using the DMP equations. All of the terms $B, \mathcal{I}, \mathcal{P}$ and $\mathcal{D}$ are forced through the use of Lagrange multipliers.

For different problems, the constraints and objectives vary. We take the competitive process as an example. In this problem, we would like to find an optimal allocation of a limited number of spreaders (budget, potentially time dependent) for process $A$, which minimizes the spreading of process $B$. These could represent a competition in a political or commercial setting. The objective function in this case is

$$\mathcal{O} = \sum_i \left( 1 - P_i^B(T) \right),$$  \hspace{1cm} (17)

where $T$ is the end of the set time window. Our goal is to maximize this objective function, thus minimizing the spread of process $B$. The resources at our disposal for seeding nodes with process $A$ are represented by the budget constraint at time zero (although more elaborate budget constraint could be accommodated as in [33] and in some of the examples that follow)

$$B_\nu = \sum_i \nu_i(0),$$  \hspace{1cm} (18)
where $\nu^i(0)$ is the deployment of seed (or fraction/probability of it) of process $A$ on node $i$. The budget constraint is forced through the Lagrange multiplier $\lambda^{Bu}$, such that

$$B = \lambda^{Bu}(B_\nu - \sum_i \nu^i(0)).$$

(19)

The fraction/probability $\nu^i$ is kept within a certain range, determined by the upper and lower bounds $\nu$ and $\nu$, respectively. Also this is enforced using a Lagrange multiplier $P = \epsilon \sum_i (\log(\nu - \nu^i(0)) + \log(\nu^i(0) - \nu))$

(20)

The term that describes the dynamics $\mathcal{D}$, introduces the dynamical equations, e.g., Eqs. (A4)-(A9) for the competitive case, using a set of Lagrange multipliers as detailed in Appendix D for the competitive case in Eq. (D4). The remaining term $\mathcal{I}$ forces the initial conditions for the dynamics as detailed in Eq. (D3).

The extremization of the Lagrangian (16) is done as follows. Variation of $\mathcal{L}$ with respect to the dual variables (Lagrange multipliers) results in the DMP equations starting from the given initial conditions, while derivation with respect to the primal variables (control and dynamic parameters) results in a second set of equations, coupling the Lagrange multipliers and the primal variable values at different times. End conditions for the forward dynamics provide the initial conditions for the backward dynamics. We solve the coupled systems of equations by forward-backward propagation, a widely used method in control and detailed in [33]. This method has a number of advantages compared to other localized optimization procedures such as gradient descent and its variants. It is simple to implement, of modest computational complexity and does not require any adjustable parameters. The forward-backward optimization provides resource (budget) values to be placed at time zero (or of at any time within the time window if we so wish) in order to optimize the objective function, e.g., that of Eq. (17).

V. NUMERICAL EXPERIMENTS

To validate and demonstrate the efficacy of the optimization method we carry out experiments on both synthetic and realistic networks. Before embarking on a large-scale application we study the performance of the derived method on a tree-like network of 30 nodes.

A. Validation of the Optimization Algorithm

To validate the DMP optimization algorithm on a problem that could be exhaustively studied and intuitively presented, we restrict ourself to a small exemplar tree-like synthetic model. Moreover, we select a small number of nodes (3) on which resource could be deployed. The objective is to maximize the spread of the two agents or to minimize the spreading of one of them (the disease control scenario) in both competitive and collaborative processes.

The 30-node network used for carrying out these experiments is presented in Fig. 3. Comparison between the DMP-based optimization algorithm and the exhaustive search is implemented in the following way: We consider a scenario where the entire budget is available at time $t = 0$. The optimization problem minimizes the spreading of process $B$ in a competitive processes through judicious budget allocation of the seeds for process $A$, i.e. we aim at minimizing $\sum_i P_i^B(T)$, where $T$ is the end time of the process.
In the experiments, resources for process $A$ were deployed on three nodes: node 0 marked as $\texttt{Controllable\_node}[0]$, node 19 marked as $\texttt{Controllable\_node}[1]$ and node 28 (not shown in the graph since it is determined by the choices for nodes 0 and 19 due to the total budget constraint of one). This choice is arbitrary and insignificant. The fixed seed for process $B$ is node 3. The objective function landscape has been explored by sampling for different parameter values as denoted by the green points in Fig. 4(a). The collaborative scenario with specific infection parameters is plotted in Fig. 4(b). A different set of controllable nodes is presented in Fig. 4(c) and Fig. 4(d) for the competitive and collaborative cases, respectively. The maximum value obtained for the objective function is contrasted with the results obtained using the DMP-based optimization procedure, marked by the red dot, in all cases. It is clear that there is good agreement between the DMP-optimal values obtained and the optima discovered through sampling.
FIG. 4. (a) Comparison between results obtained using the DMP-based optimization method and random sampling in parameter space, in the case of a competitive process. In this experiment we use a time window $T = 3$ and infection probabilities $p_A = 0.5$, $p_B = 0.5$. Sampled values for the various control parameters are marked by green points while the DMP-optimal value is marked by a red point. Nodes 0, 28 and 19 were infected by process $A$ as controllable nodes, with a total budget of one, while the fixed seed for process $B$ is node 3. The objective in this case is containing the spread of process $B$, namely maximizing $\sum_i (1 - P_i^B(T))$. (b) A similar experiment to (a), with the same conditions and objective, for the collaborative process with double infection parameters $p_{AB} = 0.8$, and $p_{BA} = 0.8$. (c) Comparing results obtained for the competitive process using DMP-based optimization and sampling of the and the collaborative process. Time window and infection parameters are as in (a), with nodes 6, 8 and 15 being the controllable nodes and the node 3 is infected by process $B$. The objective in this case is maximizing the multi-agent seeding, i.e., maximizing $\sum_i (1 - P_i^S(T))$. (d) A similar experiment to (c) for the collaborative process with double infection parameters as is (b).
B. Results on Real Networks

1. Competitive processes

We study the performance of the optimization process on networks assuming that all resources are available at time zero. Other optimization strategies where resources’ availability is time-dependent could also be considered [33] as shown later on. One of the problems in studying the efficacy of our method is the limited number of dedicated competing approaches, especially in the case of time-dependent resource deployment. We therefore compare the DMP-optimized spreading process with a uniform allocation of resources over all nodes (except seed nodes induced with process $B$), the high degree deployment strategy [39] HDA and $K$-shell decomposition [21]. Free spreading refers to an uncontrolled spread of process $B$. Blocking refers to a competing agent that is non-infectious ($p = 0$). In all cases, resources are used to contain the spreading of process $B$.

In these experiments we allocated seeds of process $B$ on $0.05N$ nodes at time zero, where $N$ is the total number of nodes in the network. The same amount of resource was allocated to the controlled competing process $A$. Infection parameters are $p = 0.2$ and $q = 0.3$. The parameter that forces the upper/lower limits of the resource variables (D2) is set to $\epsilon = 0.1$ initially and decay exponentially with iteration steps. Experimental results show that decaying $\epsilon$ in this manner leads to a improved performance, arguably since it allows one to obtain solutions which are closer to the limiting values. The optimization procedure is iterated 10 times and the best result is selected. The objective is to minimize the spreading of process $B$ and the normalized total spreading $\sum_i P_i^B(T)/N$ at time $T = 3$ is shown in Table I. The short time window used is due to the small diameter of the networks.

In competitive processes we observe that the containment of spreading process $B$ is carried out effectively by optimal seeding of spreading agents of process $A$, compared to a static blocking, HDA deployment, $K$-shell or uniform seeding. This scenario corresponds to marketing, rumor spreading/fake news and opinion setting. We also evaluate the improvement obtained for a given budget in blocking the spread, which allows one to allocate the appropriate budget for containment (e.g., addressing the spread of fake news or antivaxxing rumors by releasing verifiable information). Clearly, key factors in determining the spread are the actual infection parameters associated with the various processes, which can be obtained through data analysis. The Football network is highly connected and we speculate that this is the reason for the success of uniformly allocating the spreading agents.

When the budget allocation is done dynamically at different times, several optimization procedures can be used. For instance, consider a case where one process spreads randomly while for the other a unit budget is available per time step to be deployed optimally. Given the information at hand for each time step, one may then consider the following dynamic allocation strategies: (a) DMP greedy deploys the unit to optimize the objective function for the next time step; while (b) DMP optimal optimizes the objective function for the end-time $T$. Note that the latter represents a closed-loop optimization in the sense that updated information at each time step is available.

To demonstrate the efficacy of our method and the differences between the DMP-greedy and DMP-optimal resource deployment, we use the Football benchmark network and infect node 1 at time 0 (blue point left of the center, total budget for $B$ is 1); we then deploy optimally a budget of 1 for process $A$ at each time step $t = 1, 2, \ldots, 5$. The infection parameters used are $p_A = p_B = 0.7$. The results are shown in Fig. 5 where the heat bar represents the dominating process per node through the value $P_A^i(t) - P_B^i(t)$. Red and blue represent dominating processes $A$ and $B$, respectively. It is clear that DMP-optimal (b) is much more effective than DMP-greedy (b) in restricting the spread of process $B$ by maximizing the spread of process $A$. Numerical comparison between the two methods for the same
| Network   | Number of nodes | DMP Allocation | Uniform Allocation | K-shell | HDA   | Blocking | Free Spreading |
|-----------|-----------------|----------------|--------------------|---------|-------|----------|----------------|
| Football  | 115             | 0.5829         | 0.5674             | 0.8264  | 0.731 | 0.7860   | 0.8264         |
| Lesmis    | 77              | 0.0834         | 0.2211             | 0.1962  | 0.0990| 0.1016   | 0.3549         |
| Karate    | 35              | 0.4206         | 0.4902             | 0.5470  | 0.4212| 0.4966   | 0.5472         |
| Power     | 4941            | 0.0500         | 0.0621             | 0.0641  | 0.0632| 0.0501   | 0.0644         |
| Polbooks  | 105             | 0.2243         | 0.3733             | 0.3102  | 0.2779| 0.4316   | 0.5744         |

**TABLE I.** Comparing different allocation approaches for competitive scenario on various networks. On each network, we randomly choose $0.05N$ nodes as seeds of process $B$ and the same total budget for the competitive process $A$, where $N$ is number of nodes in the network. Infection parameters are $p = 0.2$ and $q = 0.3$. The initial parameter $\epsilon$ which forces the budget limits per node as in (D2) is initially set to 0.1 but decays exponentially with the iteration steps. The optimization procedure is iterated 10 times and the best result is selected. The objective is to minimize the spreading of process $B$; the normalized total spreading $\sum_i P_i^B(T)/N$ at time $T=3$ is shown. In all methods budgets are available at time zero. Uniform allocation assumes all the budgets are uniformly allocated to all nodes at time zero (except those infected with process $B$). In Free Spreading, no agents are allocated for process $A$. In Blocking, competitive agents are not infectious ($p = 0$), all the budget is used to contain the spreading of $B$. Best results are denoted by bold fonts.

Table and conditions are presented in **Table II**: it is clear that while DMP-greedy is successful at earlier time steps, DMP-optimal minimizes the spread of process $B$ at $T = 5$. A second example on the denser network of is given in Appendix E.

| Time | DMP optimal | DMP greedy |
|------|-------------|------------|
| $t=1$| 0.0767      | **0.0556** |
| $t=2$| 0.2157      | **0.1871** |
| $t=3$| **0.3109**  | 0.4336     |
| $t=4$| **0.3211**  | 0.5615     |
| $t=5$| **0.3211**  | 0.5628     |

**TABLE II.** DMP based resource deployment methods for the Football network under the same condition as in Fig. 5. The objective is to minimize the spreading of processes $B$, i.e. minimize the fraction of $B$ nodes at time $T=5$, $\sum_i P_i^B(T)/N$.

To demonstrate the improvement in reducing the spread of agent $B$ given a budget $b_B$ against an optimally deployed budget $b_A$ of process $A$, we plotted the ratio of infected $A$ and $B$ states against the budget ratio $b_A/b_B$. This has been done for the Lesmis network with infection probabilities $p_A = p_B = 0.5$ and time window $T = 5$; the results shown in Fig. 6(a) have been averaged over 5 instances for both uniform and optimal DMP-based deployment. It is clear from the figure that the optimal DMP-based deployment provides much better results than uniform deployment, which exhibits a linear increase. The saturation for high ratios is limited by the size of the graph.

We observed several instances in which the ratio of infected probabilities exhibits a fast transition at specific points. Figure 6(b) shows a fast transition towards a $A$-process dominated network as the ratio between the budgets allocated exceeds a certain value $b_A/b_B \approx 0.28$. The $y$-axis represents the ratio of process probabilities $\sum_i P_i^A(T)/ \sum_i P_i^B(T)$. These results clearly depend on the topology, budgets and infection rates used. The example given here is based on
FIG. 5. Football network, $p_A = p_B = 0.7$, budget for $B$ is 1, allocated at node 1 at $t = 0$, and a budget of 1 per time step is assigned optimally for process $A$. The figures represent the containment of process $B$ at different times $t = 1, 2, \ldots, 5$ due to the judicious allocation of resources of process $A$. (a) DMP-greedy; (b) DMP-optimal. The heat bar represents the dominating process per node through the value $P^i_A(t) - P^i_B(t)$. Red/Blue represent dominating processes $A/B$, respectively.

the Lesmis network with infection probabilities $p_A = 0.5$ and $p_B = 0.7$, fixed budget $b_B = 1$, time window $T = 5$ and optimal DMP-based deployment.

To evaluate the interplay between infection parameter values and budget allocated to each of the processes, we studied a competitive case on the Lesmis network, with a varying ratio between budgets ($b_B = 1, b_A = 0.5, \ldots, 4.5$) and infection probabilities ($p_B = 0.7, p_A = 0.1, \ldots, 0.7$). The points where the processes end up with equal probabilities at $T = 5$ are plotted in Fig. 6(c) for the optimized DMP deployment of resource $A$ (green line) and for uniform deployment (red line). The results are averaged over randomly chosen 5 initial position for the seed of $B$. From the figure it is clear that in this case DMP optimized deployment can effectively mitigate significantly inferior infection or budget ratios (area above the green curve); only when both ratios are very low the $B$ process dominates the network after $T$ steps (yellow area). Uniform deployment results in much inferior performance (area above the red line).
FIG. 6. Optimized competitive scenario on the Lesmis network. (a) The ratio of nodes infected by processes $A$ and $B$ after $T = 5$ time steps for a given budget ratio $b_A/b_B$. The infection probabilities used are $p_A = p_B = 0.5$. The green curve represents ratio in the case of DMP optimized deployment of the $A$ budget, while the red curve represents the uniform deployment case. (b) A fast transition in the ratio of probabilities $\frac{\sum_{i=1}^{N} P_i^A(T)}{\sum_{k=1}^{N} P_k^B(T)}$ for infection probabilities $p_A = 0.5$ and $p_B = 0.7$, and time window $T = 5$. The fixed budget for $b_B = 1$ and the budget deployment for $A$ was optimized using DMP. (c) The interplay between infection parameter values and budget allocated to each of the processes on the Lesmis network. The $x$-axis represents the between infection parameters $p_A/p_B$ and the $y$-axis the ratio between budgets $b_A/b_B$. The green line represent values for which the two processes have equal probabilities in the network at $T = 5$ when optimized DMP deployment of resource $A$ is used; the red line represents the same line for uniform deployment. Results are averaged over randomly chosen 5 initial positions for the seed of $B$.

2. Collaborative processes

Results obtained for collaborative scenarios, shown in Table B, exhibit a similar behavior, showing that collaborative processes can be optimized to spread quickly due to the supportive role played by the two processes. In this
case, some nodes \((0.05N)\) are infected by process \(B\) and we allocate a given budget of spreading process \(A\) such as that the joint spreading will be maximized and the number of non-infected nodes (in state \(S\)) minimized. This could represent, for instance, the spread of opinions on the basis of political affiliation. Also in this case the DMP-based optimization algorithm works well, with the exception of the football network where uniform spreading seems to be successful, arguably due to the same reasons as in the competitive case.

| Network  | Number of nodes | DMP Allocation | Uniform Allocation | K-shell | HDA     | Free Spreading |
|----------|-----------------|----------------|--------------------|---------|---------|----------------|
| Football | 115             | 0.0536         | 0.0582             | 0.1736  | 0.1543  | 0.1735         |
| Lesmis   | 77              | 0.2222         | 0.3832             | 0.3151  | 0.3051  | 0.6451         |
| Karate   | 35              | 0.2771         | 0.3743             | 0.4527  | 0.2481  | 0.4529         |
| Power    | 4941            | 0.7652         | 0.7930             | 0.8434  | 0.9024  | 0.9355         |
| Polbooks | 105             | 0.1524         | 0.2065             | 0.3347  | 0.2204  | 0.4255         |

TABLE III. Comparing different allocation approaches for collaborative processes on various networks. On each network, we randomly select \(0.05N\) nodes as seeds of process \(B\) and the same total budget for the competitive process \(A\) to be optimally allocated, where \(N\) is total number of nodes in the network. The infection parameters chosen are \(p_A = 0.2, p_B = 0.3, p_{AB} = 0.4\) and \(p_{BA} = 0.5\). The initial parameter \(\epsilon\) which forces the budget limits per node as in (D2) is initially set to 0.1 but decays exponentially with the iteration steps. The optimization procedure is iterated 10 times and the best result is selected. The objective function in this case is to maximize the spreading of processes \(A\) and \(B\), i.e., minimize the fraction of susceptible nodes \(\sum_i P_i^A(T)/N\) at time \(T = 3\). In DMP allocation-based optimization, we assume the seeding budget for \(B\) is available at time zero. Uniform allocation assumes that the budget for \(A\) is allocated uniformly to all free nodes initially. HDA deployment [39] and K-shell [21] seeding are used as before; in Free Spreading, no collaborative spreading budget \(A\) is allocated. Lowest fraction of susceptible nodes is denoted by bold fonts.

The final set of experiments we carried out relates to the vaccination policy in collaborative spreading processes. In this case, two collaborative processes \(A\) and \(B\) spread throughout the system and the task is to minimize the spread through a vaccination process to one of the two, say \(B\). The role of vaccination in this case is in reducing the infection probability by the level of vaccination deployed. For instance, we adopt a simplistic model whereby if the vaccine deployed at a given node is \(b\), its probability of being infected will reduce to \(p_B - b\). Similarly, the infection parameter \(p_{BA}\) will also decrease by the same amount to \(p_{BA} - b\). Other infection models can be easily accommodated. The optimization task is in the deployment of a given vaccination budget such that the fraction of non-infected sites \(\sum_i P_i^A(T)/N\) at time \(T\), is maximized.

Results of the experiment demonstrate that the DMP-based optimization shows good performance.
Generally, for the different network topologies and size, DMP-based optimization appears to be more effective than other methods. The main cases where it does not offer the best result are in complex networks with bounded connectivity fluctuation, where it has been shown that uniformly applied immunization strategies are highly effective [39].

To demonstrate the efficacy of DMP-optimal resource deployment for vaccination in the case of a collaborative spreading process we use the main road network of England [43]. Animal epidemics often spreads through the transport of infected livestock on the road network (as was the case in the 2001 Foot and Mouth epidemic in the UK). We consider the spread of highly infectious coupled processes through the road network starting from the areas of Greater London (node 10, process A) and Leeds (node 11, process B); the budget for both A and B is one unit and infection parameters $p_A = p_B = 0.2$, $p_{BA} = p_{AB} = 0.99$. The process is runs for 20 steps. The vaccination budget is one unit per time step and is effective to suppress process B only, while ineffective for process A. The results shown in Fig. 7 demonstrate the efficacy of the DMP-optimal vaccination strategy aimed at minimizing $\sum_i (1 - P_i^i(t))$ (b) in contrast to the free spreading of both infections (a). Blue/red represent uninfected/infected states, respectively; more specifically, the node color represents $1 - P_i^i(t)$, where red and blue correspond to 1 and 0, respectively. As we can see in Fig. 7, the infection spread around London remains the same, with or without the deployment of vaccine as it is mainly infected by process A, and hence the vaccination has no effect, while spread emanating from the Leeds area is effectively blocked.

| Network    | Number of nodes | DMP Allocation | Uniformed Allocation | Free Spreading |
|------------|-----------------|----------------|----------------------|----------------|
| Football   | 115             | 0.1887         | 0.0607               | 0.0294         |
| Lesmis     | 77              | 0.9455         | 0.5519               | 0.4941         |
| Karate     | 35              | 0.4676         | 0.3491               | 0.3123         |
| Power      | 4941            | 0.8891         | 0.8650               | 0.8356         |
| Polbooks   | 105             | 0.5483         | 0.3071               | 0.2354         |

TABLE IV. Different vaccine-allocation policies in a collaborative process on difference benchmark networks. On each network, we randomly choose 0.05$N$ of the nodes as seeds for process $B$ and 0.01$N$ nodes as seeds for process $A$; the total vaccination budget is 0.05$N$, where $N$ is number of nodes in the network. Infection parameters are arbitrarily set to $p_A = 0.4$, $p_B = 0.4$, $p_{AB} = 0.9$ and $p_{BA} = 0.9$. The parameter $\epsilon$ in (D2) is set to 0.01. The optimization procedure is iterated 10 times and the best result is selected. The objective is to minimize the spreading of processes $A$ and $B$, i.e., maximize the fraction of susceptible nodes $\sum_i P_i^i(T)/N$ at the end of the process (time $T = 3$). In DMP-based optimal allocation, we assume that all budgets are available at time zero. Uniform Allocation assumes all budgets to be uniformly allocated to all nodes at time zero. In Free Spreading, no vaccine is allocated. A detailed description of the vaccine allocation problem can be found in Appendix G.
FIG. 7. England road network, the collaborative spread of highly infectious coupled processes. We consider the spread of highly infectious coupled processes through the road network starting from the areas of Greater London (node 10, process $A$) and Leeds (node 11, process $B$); the budget for both $A$ and $B$ is one unit and infection parameters $p_A = p_B = 0.2$, $p_{BA} = p_{AB} = 0.99$. The process is runs for 20 steps. The vaccination budget is one unit per time step and is effective to suppress process $B$ only, while ineffective for process $A$. The node color represents the value of $1 - P_i(t)$, where red and blue correspond to 1 and 0, respectively (the heat bar is similar to that of Fig. 5). (a) Free spread of the epidemics with no vaccination. (b) A vaccination budget of one unit per time step against process $B$ is deployed at each time step using the DMP-optimal algorithm. We see that in (b) that the infection in the of London area spreads unhindered in both (a) and (b) as it is mainly infected by process $A$, while spread emanating from the Leeds area is effectively blocked.
VI. SUMMARY AND FUTURE WORK

Competition and collaboration between spreading processes are prevalent on social networks, health-related processes and many other scenarios. The cases of competitive and collaborative spreading processes are investigated using the probabilistic DMP framework and provide a good description of the spreading dynamics, as validated on both synthetic and real networks.

We employ an optimization framework that incorporates the DMP dynamics to deploy limited seeding resource in order to maximize its spread with respect to an adversarial process, thus containing the spread of the latter, as well as to maximize the joint spread in a collaborative process scenario. The optimization algorithm is tested on a range of small-scale and real networks and shows excellent performance with respect to the existing alternatives of high-degree allocation, K-shell based approaches, free spreading and uniform budget allocation. Another scenario we explored is that of optimized vaccination in the case of collaborative spreading processes, where vaccination is effective with respect to one of them only but helps to reduce the spread of both due to the interaction between them.

The suggested framework is highly adaptive and can accommodate: targeted spreading, where only specific vertices are targeted and at specific times (critical votes, time-sensitive campaigns); temporal deployment of resources, either of the process of interest or of an adversarial process, where resources are available at different times within the time window (e.g., due to limited production or shipment restrictions); different vaccination policies and more complex collaborative scenarios.

The application of this method for process containment (e.g. the spread of fake news, anti-vaxxing messages), for effective information dissemination (e.g. marketing or opinion-setting material) and for suggesting vaccination policy to contain the spread of collaborative health conditions, are promising applications that should be explored.

ACKNOWLEDGMENTS

We would like to thank Andrey Lokhov for helpful discussions and comments on the manuscript and Ho Fai Po for compiling the UK road network data. Support by the Leverhulme trust (RPG-2018-092) and the EPSRC Programme Grant TRANSNET (EP/R035342/1) is acknowledged.

[1] Wikipedia contributors, "Wannacry ransomware attack — Wikipedia, the free encyclopedia" (2018), [Online; accessed 13-September-2018], URL https://en.wikipedia.org/w/index.php?title=WannaCry_ransomware_attack&oldid=858915262.

[2] Wikipedia contributors, "African swine fever virus — Wikipedia, the free encyclopedia" (2018), [Online; accessed 13-September-2018], URL https://en.wikipedia.org/w/index.php?title=African_swine_fever_virus&oldid=858939031.

[3] Global Citizen, "Everything you need to know about the anti-vaxxer movement" (2017), [Online; accessed 12-March-2019], URL https://www.globalcitizen.org/en/content/everything-you-need-to-know-about-the-anti-vaxxer/.
[29] D. Kempe, J. Kleinberg, and É. Tardos, in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (ACM, 2003), pp. 137–146.

[30] W. Chen, L. V. Lakshmanan, and C. Castillo, *Synthesis Lectures on Data Management* 5, 1 (2013).

[31] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, in *Advances in Neural Information Processing Systems* (2013), pp. 3147–3155.

[32] C. Nowzari, V. M. Preciado, and G. J. Pappas, *IEEE Control Systems* 36, 26 (2016).

[33] A. Y. Lokhov and D. Saad, *Proceedings of the National Academy of Sciences of the United States of America* 114, E8138 (2017), ISSN 1091-6490, URL http://www.ncbi.nlm.nih.gov/pubmed/28900013 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5625886.

[34] W. Cai, L. Chen, F. Ghanbarnejad, and P. Grassberger, *Nature Physics* 11, 936 (2015), ISSN 1745-2473, URL http://www.nature.com/doifinder/10.1038/nphys3457.

[35] J. C. Miller, *Phys. Rev. E* 87, 060801(R) (2013), URL https://link.aps.org/doi/10.1103/PhysRevE.87.060801.

[36] M. E. J. Newman and C. R. Ferrario, *PLOS ONE* 8, 1 (2013), URL https://doi.org/10.1371/journal.pone.0071321.

[37] M. Girvan and M. E. J. Newman, *Proceedings of the National Academy of Sciences of the United States of America* 99, 7821 (2002), ISSN 0027-8424, URL http://www.ncbi.nlm.nih.gov/pubmed/12060727 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC122977.

[38] V. Krebs (2019), [Online; accessed 9-April-2019], URL http://www.orgnet.com/.

[39] R. Pastor-Satorras and A. Vespignani, *Phys. Rev. E* 65, 036104 (2002), URL https://link.aps.org/doi/10.1103/PhysRevE.65.036104.

[40] D. E. Knuth, *The Stanford GraphBase: a platform for combinatorial computing* (ACM Press, New York, N.Y., 1993), ISBN 0201542757, URL https://searchworks.stanford.edu/view/2789522.

[41] W. W. Zachary, *Journal of Anthropological Research* 33, 452 (1977), ISSN 0091-7710, URL https://www.journals.uchicago.edu/doi/10.1086/jar.33.4.3629752.

[42] D. J. Watts and S. H. Strogatz, *Nature* 393, 440 (1998), ISSN 0028-0836, URL http://www.nature.com/articles/30918.

[43] Highways England, *Highways agency network journey time and traffic flow data* (2019), [Online; accessed 5-April-2019], URL https://data.gov.uk/dataset/dc18f7d5-2669-490f-b2b5-77f27ec133ad/highways-agency-network-journey-time-and-traffic-flow-data.

[44] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, *Physical Review E* 90, 012801 (2014), ISSN 1539-3755, URL https://link.aps.org/doi/10.1103/PhysRevE.90.012801.

[45] W. D. Nooy, A. Mrvar, and V. Batagelj, *Exploratory Social Network Analysis with Pajek*, 0521174805, 9780521174800 (Cambridge University Press, New York, NY, USA, 2011).

[46] H. Zhang (2018), URL http://studentwork.prattsi.org/infovis/author/hzhang14pratt-edu/.

[47] Wikipedia contributors, *Bcg vaccine — Wikipedia, the free encyclopedia* (2018), [Online; accessed 13-September-2018], URL https://en.wikipedia.org/w/index.php?title=BCG_vaccine&oldid=856892624.
Appendix A: Message Passing Equations-Mutually Exclusive Competitive Processes

As explained in [44], $\theta_A^{i \rightarrow j}(t)$ (and $\theta_B^{i \rightarrow j}(t)$) represent the probability that no infection message $A$ (and $B$) have been passed from node $i$ to node $j$ until time $t$ on the cavity graph $D_j$:

$$
\theta^{i \rightarrow j}(t) = \text{Prob}^{D_j} \left[ \sum_{t'=0}^{t} d^{i \rightarrow j}(t') = 0 \right] \tag{A1}
$$

where the message $d^{i \rightarrow j}(t)$ is defined as:

$$
d^{i \rightarrow j}(t) = 1 \left[ p_I \delta_{q_i(t-1),I} \right] \tag{A2}
$$

where $q_i(t)$ represents the state of node $i$ at time $t$ and $I$ means the ”infected” state, which is state $A$ or $B$ in this competitive scenario, and $p_I$ means the infection probabilities, respectively. Meanwhile, $\phi^{i \rightarrow j}(t)$ is defined as:

$$
\phi^{i \rightarrow j}(t) = \text{Prob}^{D_j} \left[ \sum_{t'=0}^{t} d^{i \rightarrow j}(t') = 0, q_i(t) = I \right] \tag{A3}
$$

representing the probability that no infection message has been passed until time $t - 1$, while node $i$ is infected at time $t$. The probabilistic message $P_{\theta}^{i \rightarrow j}(t)$ means the renormalized probabilities of node $i$ on a specific state $\sigma$ ($\sigma = S, A, B$) at time $t$, while $\hat{P}_{\theta}^{i \rightarrow j}(t)$ means the probabilities before renormalization.

We use similar notations used in [44], where detailed ideas of deriving message passing equation can be found. The details of equations are as follows:

The change for $\theta^{i \rightarrow j}(t)$ can only result from node $i$ passing an infection message to node $j$ at time $t$. Therefore we have:

$$
\theta_A^{i \rightarrow j}(t) - \theta_A^{i \rightarrow j}(t-1) = -p \phi_A^{i \rightarrow j}(t-1) \\
\theta_B^{i \rightarrow j}(t) - \theta_B^{i \rightarrow j}(t-1) = -q \phi_A^{i \rightarrow j}(t-1) \tag{A4}
$$

The change for $\phi^{i \rightarrow j}(t)$ may result either from node $i$ sending an infection message to node $j$ at time $t$, or when node $i$ is infected at time $t$. Therefore we have:

$$
\phi_A^{i \rightarrow j}(t) = (1 - p)\phi_A^{i \rightarrow j}(t-1) + \left[ P_A^{i \rightarrow j}(t) - P_A^{i \rightarrow j}(t-1) \right] \\
\phi_B^{i \rightarrow j}(t) = (1 - q)\phi_B^{i \rightarrow j}(t-1) + \left[ P_B^{i \rightarrow j}(t) - P_B^{i \rightarrow j}(t-1) \right] \tag{A5}
$$

Since the iterative processes includes a renormalization procedure the message $\hat{P}_S^{i \rightarrow j}(t)$ cannot be obtained directly from the initial conditions and $\theta^{i \rightarrow j}(t)$, as in [44]. Instead, the probability can be each step at a time by calculating the conditional probability that node $i$ was at state $S$ at time $t - 1$ and no message, $A$ or $B$, has been passed to node $i$ at time $t$, which indicates a probability. Therefore the iterative equation for $\hat{P}_S^{i \rightarrow j}(t)$ takes the form:

$$
\hat{P}_S^{i \rightarrow j}(t) = P_S^{i \rightarrow j}(t-1) \prod_{k \in \partial \setminus \{j\}} \frac{\theta_A^{k \rightarrow i}(t) \theta_B^{k \rightarrow i}(t)}{\theta_A^{k \rightarrow i}(t-1) \theta_B^{k \rightarrow i}(t-1)} \tag{A6}
$$

The next step is to obtain the change of $\hat{P}_A^{i \rightarrow j}$, when at least one of the neighbors of node $i$ passes an infection message $A$ to it while no $B$ messages are received. Additionally, node $i$ should be in state $S$ at time $t - 1$, i.e., neither messages
A nor B have been passed to node i before t – 1. The probability of node i being infected by A or B becomes to a conditional probability, leading to the form \( \prod_{k \in \partial i \setminus j} \left( 1 - \frac{q \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \) and the change of \( \widehat{P}_{i \rightarrow j}(t) \) becomes:

\[
\widehat{P}_{A \rightarrow j}(t) = P_{S \rightarrow j}(t-1) \left[ \prod_{k \in \partial i \setminus j} \left( 1 - \frac{q \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \left( 1 - \prod_{k \in \partial i \setminus j} \left( 1 - \frac{p \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \right) \right] + P_{A \rightarrow j}(t-1)
\]

\[
\widehat{P}_{B \rightarrow j}(t) = P_{S \rightarrow j}(t-1) \left[ \prod_{k \in \partial i \setminus j} \left( 1 - \frac{p \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \left( 1 - \prod_{k \in \partial i \setminus j} \left( 1 - \frac{q \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \right) \right] + P_{B \rightarrow j}(t-1)
\] (A7)

We then compute the renormalized probabilities for the three states \( P_{S/A/B \rightarrow j}(t) \):

\[
P_{S/A/B \rightarrow j}(t) = \frac{\widehat{P}_{S/A/B \rightarrow j}(t)}{\widehat{P}_{S \rightarrow j}(t) + \widehat{P}_{A \rightarrow j}(t) + \widehat{P}_{B \rightarrow j}(t)}
\] (A8)

The posterior estimates are then calculated iteratively on a complete graph:

\[
\widehat{P}_{S \rightarrow i}(t) = P_{S \rightarrow i}(t-1) \left[ \prod_{k \in \partial i} \frac{\theta_{k \rightarrow i}(t) \theta_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t) \theta_{k \rightarrow i}(t-1)} \right] + P_{i \rightarrow j}(t-1)
\]

\[
\widehat{P}_{A \rightarrow i}(t) = P_{S \rightarrow i}(t-1) \left[ \prod_{k \in \partial i} \left( 1 - \frac{q \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \left( 1 - \prod_{k \in \partial i} \left( 1 - \frac{p \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \right) \right] + P_{A \rightarrow i}(t-1)
\]

\[
\widehat{P}_{B \rightarrow i}(t) = P_{S \rightarrow i}(t-1) \left[ \prod_{k \in \partial i} \left( 1 - \frac{p \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \left( 1 - \prod_{k \in \partial i} \left( 1 - \frac{q \phi_{k \rightarrow i}(t-1)}{\theta_{k \rightarrow i}(t-1)} \right) \right) \right] + P_{B \rightarrow i}(t-1)
\]

\[
P_{S/A/B \rightarrow i}(t) = \frac{\widehat{P}_{S/A/B \rightarrow i}(t)}{\widehat{P}_{S \rightarrow i}(t) + \widehat{P}_{A \rightarrow i}(t) + \widehat{P}_{B \rightarrow i}(t)}
\] (A9)

Appendix B: Message Passing Equations-Collaborative Processes

In the collaborative case, we regard the state AB as a coexistence of processes A and B. In other words, in our notation, \( P_{i \rightarrow j}(t) = P_{A \rightarrow B \rightarrow j}(t) + P_{A \rightarrow j}(t) \), where \( P_{i \rightarrow j}(t) \) represents the probability of node i being a pure A state (no B infection) at time t. Therefore, the normalization relation \( P_{i \rightarrow j}(t) + P_{i \rightarrow j}(t) + P_{i \rightarrow j}(t) + P_{i \rightarrow j}(t) = 1 \) does not hold anymore; instead, the relation becomes \( P_{A \rightarrow j}(t) + P_{B \rightarrow j}(t) + P_{S \rightarrow j}(t) + P_{i \rightarrow j}(t) = 1 \). We use this notation since in studying the spreading of disease A, for instance, state A and state AB are both a sources of infection. The detailed equations are as follows (subindex A/B refer to either of the two processes as the corresponding two equations are symmetric):

\[
\theta_{A \rightarrow B / B \rightarrow A \rightarrow j}(t) = P_{A \rightarrow j}(t) \prod_{k \in \partial i \setminus j} \theta_{A \rightarrow i}(t) \theta_{B \rightarrow i}(t)
\]

\[
\phi_{A \rightarrow B / B \rightarrow A \rightarrow j}(t) = P_{A \rightarrow j}(t) \prod_{k \in \partial i \setminus j} \phi_{A \rightarrow i}(t) \phi_{B \rightarrow i}(t) + P_{A \rightarrow j}(t) - P_{A \rightarrow j}(t-1)
\] (B1)
To derive the dynamics of the messages we should consider both the infection through a pure state and the state including both infections as described below. For instance, in Eq. (B2), the second term on the right hand side of the equation represents the increase in $P_{A/B}$ by nodes in state $S$ at $t−1$ that become infected via the processes $S \rightarrow A/B$ and $S \rightarrow AB$. Since the infection process of $A$ and $B$ is independent at the same time step, we only care about messages of process $A$ and ignore process $B$ (and the other way around in the symmetric equation). The third term represent the process $B \rightarrow AB$.

\[
P_{i \rightarrow j}^{A/B}(t) = P_{i \rightarrow j}^{A/B}(t−1) + P_{i \rightarrow j}^{S}(t−1) \left( 1 - \prod_{k \in \partial_i \setminus j} \left( 1 - p_{A/B}^k \phi_{A/B}^k(t−1) \theta_{A/B}^k(t−1) \right) \right)
\]

\[
+ P_{(B/A)_*}^{i \rightarrow j}(t−1) \left( 1 - \prod_{k \in \partial_i \setminus j} \left( 1 - \frac{p_{AB}^k \phi_{A/B}^k(t−1)}{\theta_{A/B}^k(t−1)} \right) \right)
\]

and the posterior estimates,

\[
P_{i}^{A/B}(t) = P_{i}^{A/B}(t−1) + P_{i}^{S}(t−1) \left( 1 - \prod_{k \in \partial_i} \left( 1 - \frac{p_{A/B}^k \phi_{A/B}^k(t−1)}{\theta_{A/B}^k(t−1)} \right) \right)
\]

\[
+ P_{(B/A)_*}^{i}(t−1) \left( 1 - \prod_{k \in \partial_i} \left( 1 - \frac{p_{AB}^k \phi_{A/B}^k(t−1)}{\theta_{A/B}^k(t−1)} \right) \right),
\]

where the messages obey

\[
P_{AB}^{i \rightarrow j}(t) = P_{A}^{i \rightarrow j}(t) + P_{B}^{i \rightarrow j}(t) + P_{S}^{i \rightarrow j}(t) - 1
\]

\[
P_{(A/B)_*}^{i \rightarrow j}(t) = P_{A/B}^{i \rightarrow j}(t) - P_{AB}^{i \rightarrow j}(t)
\]

and similarly for the posterior estimate

\[
P_{AB}^{i}(t) = P_{A}^{i}(t) + P_{B}^{i}(t) + P_{S}^{i}(t) - 1
\]

\[
P_{(A/B)_*}^{i}(t) = P_{A/B}^{i}(t) - P_{AB}^{i}(t)
\]

\[
P_{S}^{i}(t) = P_{S}^{i}(0) \prod_{k \in \partial_i} \theta_{A}^{k \rightarrow i}(t) \theta_{B}^{k \rightarrow i}(t)
\]

Appendix C: Inference in Collaborative Spreading

Similar experiments were run for a collaborative process on the toy tree network, graphs with loops and the benchmark Football network. The results are shown in Fig. 8 for the various cases.
FIG. 8. Comparison of DMP-based results and Monte Carlo simulations for collaborative processes. For each of the graphs $10^3$ samples have been used per node (about $10^5$ samples in total) and both mean values and error-bars are shown (unless they are smaller than the symbol size). (a) For a collaborative process on a tree-like network using the parameters $p_A = 0.3$, $p_B = 0.7$, $p_{AB} = 0.6$ and $p_{BA} = 0.6$, observing $P^{3}_{S}(t)$ (the node to monitor has been selected arbitrarily). Process A was seeded at node 7 and process B was seeded at node 2. (b) The same as in (a) but observing $P^{3}_{B}(t)$. (c) Comparing results for the network with loops using the same parameters as in (a), observing $P^{3}_{S}(t)$ and (d) $P^{3}_{A}(t)$. (e) Comparing results obtained from the DMP method and Monte Carlo simulations of a collaborative process on the Football network. The parameters used are $p_A = 0.1$, $p_B = 0.2$, $p_{AB} = 0.3$ and $p_{BA} = 0.4$. The probabilities represent observations of node 2, while process A was seeded at nodes 3 and 4, and process B at nodes 0 and 1 (there is no significance to any of these choices).
The experimental results illustrate that DMP-based modeling is very accurate on tree-like networks. It is less accurate on small loopy graphs at longer times, as expected, due to the small loops that violate the cavity method’s assumption (the specific 10-nodes network used includes 2 small loops). This effect is suppressed to some extent on larger networks where loops are typically longer as demonstrated on the real benchmark network examples. We conclude that the DMP method provides a good description in the case of competitive/cooperative spreading processes.

Appendix D: Optimization of Mutually Exclusive Competitive Processes

For competitive processes we study two optimization problems, the multi-agent seeding problem and disease containment, the only difference between the two is the objective function. For disease containment the objective function is \( O = \sum_i (1 - P_i^B(T)) \) and for the multi-agent seeding problem \( O = \sum_i (1 - P_i^S(T)) \). The budget constraint enforced by the Lagrange multiplier \( \lambda^{Bu} \) is:

\[
B = \lambda^{Bu} \left( B_\nu - \sum_i \nu^i(0) \right) \tag{D1}
\]

where permitted \( \nu < \nu^i < \bar{\nu} \) value restrictions are enforced by the term

\[
P = \varepsilon \sum_i \left( \log(\bar{\nu} - \nu^i(0)) + \log(\nu^i(0) - \nu) \right). \tag{D2}
\]

In this case, the restrictions used are \( \bar{\nu} = 1 \) and \( \nu = 0 \).

Initial conditions (\( I \) in Eq.(16)) are forced through a set of Lagrange multipliers \( \lambda \) for the various parameter values

\[
I = \sum_i \lambda_i^A(0) \left( \tilde{P}_i^A(0) - \nu^i(0) \right) \left( 1 - \delta_{\sigma_i(0),B} \right) + \sum_i \lambda_i^A(0) \left( P_{A,i}^i(0) - \nu^i(0) \left( 1 - \delta_{\sigma_i(0),B} \right) \right) + \sum_i \lambda_i^B(i) \left( \tilde{P}_i^B(0) - \delta_{\sigma_i(0),B} \right) + \sum_i \lambda_i^B(i) \left( P_{B,i}^i(0) - \delta_{\sigma_i(0),B} \right) + \sum_i \lambda_i^S(i) \left( \tilde{P}_i^S(0) - 1 + \nu^i(0) \left( 1 - \delta_{\sigma_i(0),B} \right) + \delta_{\sigma_i(0),B} \right) + \sum_i \lambda_i^{\theta_B,0}(0) \left( \theta_{B,I}^{i \rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^{\theta_A,0}(0) \left( \phi_{B,I}^{i \rightarrow j}(0) - \nu^i(0) \left( 1 - \delta_{\sigma_i(0),B} \right) \right) + \sum_{ij} \lambda_{ij}^{\theta_B,0}(0) \left( \theta_{B,I}^{i \rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^{\theta_A,0}(0) \left( \phi_{A,I}^{i \rightarrow j}(0) - \nu^i(0) \left( 1 - \delta_{\sigma_i(0),B} \right) \right) + \sum_{ij} \lambda_{ij}^{\theta_B,0}(0) \left( \theta_{B,I}^{i \rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^{\theta_A,0}(0) \left( \phi_{A,I}^{i \rightarrow j}(0) - \nu^i(0) \left( 1 - \delta_{\sigma_i(0),B} \right) \right) + \sum_{ij} \lambda_{ij}^{\theta_B,0}(0) \left( \theta_{B,I}^{i \rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^{\theta_A,0}(0) \left( \phi_{A,I}^{i \rightarrow j}(0) - \nu^i(0) \left( 1 - \delta_{\sigma_i(0),B} \right) \right) + \sum_{ij} \lambda_{ij}^{\theta_B,0}(0) \left( \theta_{B,I}^{i \rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^{\theta_A,0}(0) \left( \phi_{A,I}^{i \rightarrow j}(0) - \nu^i(0) \left( 1 - \delta_{\sigma_i(0),B} \right) \right).
\]
The DMP equations for the dynamics (\(D\) in Eq.(16)) are forced through a set of Lagrange multipliers

\[
D = \sum_{ij}^{T-1} \lambda_{ij}^{A}(t+1)[\theta_{A}^{i\rightarrow j}(t+1) - \theta_{A}^{i\rightarrow j}(t) + p\phi_{A}^{i\rightarrow j}(t)]
\]

\[
+ \sum_{ij}^{T-1} \lambda_{ij}^{B}(t+1)[\theta_{B}^{i\rightarrow j}(t+1) - \theta_{B}^{i\rightarrow j}(t) + q\phi_{B}^{i\rightarrow j}(t)]
\]

\[
+ \sum_{ij}^{T-1} \lambda_{ij}^{A}(t+1)[\phi_{A}^{i\rightarrow j}(t+1) + (p-1)\phi_{A}^{i\rightarrow j}(t) - [P_{A}^{i\rightarrow j}(t+1) - P_{A}^{i\rightarrow j}(t)]
\]

\[
+ \sum_{ij}^{T-1} \lambda_{ij}^{B}(t+1)[\phi_{B}^{i\rightarrow j}(t+1) + (q-1)\phi_{B}^{i\rightarrow j}(t) - [P_{B}^{i\rightarrow j}(t+1) - P_{B}^{i\rightarrow j}(t)]
\]

\[
+ \sum_{ij}^{T-1} \lambda_{ij}^{S}(t+1) \left[ P_{S}^{i\rightarrow j}(t+1) - P_{S}^{i\rightarrow j}(t) \prod_{k \in \partial i \cup j}^{\lambda_{k}^{A}(t+1)} \frac{\theta_{A}^{k\rightarrow i}(t+1)\theta_{B}^{k\rightarrow i}(t+1)}{\theta_{A}^{k\rightarrow i}(t)\theta_{B}^{k\rightarrow i}(t)} \right]
\]

\[
+ \sum_{ij}^{T-1} \lambda_{ij}^{S}(t+1) \left[ \frac{\hat{P}_{S}^{i\rightarrow j}(t+1)}{\hat{P}_{S}^{i\rightarrow j}(t+1) + \hat{P}_{A}^{i\rightarrow j}(t+1) + \hat{P}_{B}^{i\rightarrow j}(t+1)} \right]
\]

\[
+ \sum_{ij}^{T-1} \lambda_{ij}^{S}(t+1) \left[ \frac{\hat{P}_{A}^{i\rightarrow j}(t+1)}{\hat{P}_{A}^{i\rightarrow j}(t+1) + \hat{P}_{S}^{i\rightarrow j}(t+1) + \hat{P}_{B}^{i\rightarrow j}(t+1)} \right]
\]

\[
+ \sum_{ij}^{T-1} \lambda_{ij}^{S}(t+1) \left[ \frac{\hat{P}_{B}^{i\rightarrow j}(t+1)}{\hat{P}_{B}^{i\rightarrow j}(t+1) + \hat{P}_{A}^{i\rightarrow j}(t+1) + \hat{P}_{B}^{i\rightarrow j}(t+1)} \right]
\]

\[
(D4)
\]
Derivatives with respect to the dynamics parameters give rise to the optimization (dynamical) equations for the Lagrange multipliers. The case described here is that of the containment problem (minimizing spread of an adversarial process).

\[
\frac{\partial L}{\partial \hat{P}_S^{i \rightarrow j}(t)} = \lambda^S_{ij}(t) + 1 \quad [t \neq 0] \left[ -\lambda^S_{ij}(t) \frac{\hat{P}_A^{i \rightarrow j}(t) + \hat{P}_B^{i \rightarrow j}(t)}{(\hat{P}_S^{i \rightarrow j}(t) + \hat{P}_A^{i \rightarrow j}(t) + \hat{P}_B^{i \rightarrow j}(t))^2} + \lambda^A_{ij}(t) \frac{\hat{P}_A^{i \rightarrow j}(t)}{(\hat{P}_S^{i \rightarrow j}(t) + \hat{P}_A^{i \rightarrow j}(t) + \hat{P}_B^{i \rightarrow j}(t))^2} \right]
\]

\[
\frac{\partial L}{\partial \hat{P}_S^{i}(t)} = \lambda^S_{i}(t) + 1 \quad [t \neq 0] \left[ -\lambda^S_{i}(t) \frac{\hat{P}_A^{i}(t) + \hat{P}_B^{i}(t)}{(\hat{P}_S^{i}(t) + \hat{P}_A^{i}(t) + \hat{P}_B^{i}(t))^2} + \lambda^A_{i}(t) \frac{\hat{P}_A^{i}(t)}{(\hat{P}_S^{i}(t) + \hat{P}_A^{i}(t) + \hat{P}_B^{i}(t))^2} \right]
\]

\[
\frac{\partial L}{\partial \hat{P}_S^{i \rightarrow j}(t)} = \left[ -\lambda^S_{ij}(t+1) \prod_{k \in \partial_i \setminus j} \frac{\theta_A^{k \rightarrow i}(t+1) \theta_B^{k \rightarrow i}(t+1)}{\theta_A^{k \rightarrow i}(t) \theta_B^{k \rightarrow i}(t)} 
- \lambda^A_{ij}(t+1) \prod_{k \in \partial_i \setminus j} \left( 1 - \prod_{k \in \partial_i \setminus j} \left( 1 - \frac{q_B^{k \rightarrow i}(t)}{\theta_B^{k \rightarrow i}(t)} \right) \right) \right] 1 \quad [t \neq T] + \lambda^S_{ij}(t)
\]

\[
\frac{\partial L}{\partial \hat{P}_A^{i}(t)} = \left[ \lambda^{A_i}(t+1) - \lambda^{A_j}(t+1) \right] 1 \quad [t \neq T] + \lambda^A_{i}(t) - 1 \quad [t \neq 0] \left[ \lambda^{A_i}(t) \right]
\]

\[
\frac{\partial L}{\partial \hat{P}_B^{i}(t)} = \left[ -\lambda^{A_i}(t+1) \right] 1 \quad [t \neq T] + \lambda^A_{i}(t)
\]

\[
\frac{\partial L}{\partial \hat{P}_B^{i \rightarrow j}(t)} = \lambda^B_{ij}(t) + 1 \quad [t \neq 0] \left[ -\lambda^B_{ij}(t) \frac{\hat{P}_A^{i \rightarrow j}(t) + \hat{P}_B^{i \rightarrow j}(t)}{(\hat{P}_S^{i \rightarrow j}(t) + \hat{P}_A^{i \rightarrow j}(t) + \hat{P}_B^{i \rightarrow j}(t))^2} + \lambda^B_{ij}(t) \frac{\hat{P}_B^{i \rightarrow j}(t)}{(\hat{P}_S^{i \rightarrow j}(t) + \hat{P}_A^{i \rightarrow j}(t) + \hat{P}_B^{i \rightarrow j}(t))^2} \right]
\]

\[
\frac{\partial L}{\partial \hat{P}_B^{i}(t)} = \lambda^B_{i}(t) + 1 \quad [t \neq 0] \left[ -\lambda^B_{i}(t) \frac{\hat{P}_A^{i}(t) + \hat{P}_B^{i}(t)}{(\hat{P}_S^{i}(t) + \hat{P}_A^{i}(t) + \hat{P}_B^{i}(t))^2} + \lambda^B_{i}(t) \frac{\hat{P}_B^{i}(t)}{(\hat{P}_S^{i}(t) + \hat{P}_A^{i}(t) + \hat{P}_B^{i}(t))^2} \right]
\]

\[
(D5)
\]
\[
\frac{\partial L}{\partial \theta_A^{ij}(t)} = \lambda_{ij}^A(t) - 1[t \neq T] \left( \lambda_{ij}^A(t+1) \right)
\]

\[
+ \sum_{a \in \partial j \setminus i} \lambda_{ja}^S(t+1)P^j_{S}(t) \frac{1}{\theta_A^{ij}(t)} \prod_{l \in \partial j \setminus a} \frac{\theta_A^{ij}(t+1)\theta_B^{ij}(t+1)}{\theta_A^{ij}(t)\theta_B^{ij}(t)} 1[t \neq T]
\]

\[
- \sum_{a \in \partial j \setminus i} \lambda_{ja}^S(t)P^j_{S}(t-1) \frac{1}{\theta_A^{ij}(t)} \prod_{l \in \partial j \setminus a} \frac{\theta_A^{ij}(t)\theta_B^{ij}(t)}{\theta_A^{ij}(t-1)\theta_B^{ij}(t-1)} 1[t \neq 0]
\]

\[
+ \lambda_{j}^S(t+1)P^j_{S}(t) \frac{1}{\theta_A^{ij}(t)} \prod_{l \in \partial j} \frac{\theta_A^{ij}(t+1)\theta_B^{ij}(t+1)}{\theta_A^{ij}(t)\theta_B^{ij}(t)} 1[t \neq T]
\]

\[
- \lambda_{j}^S(t)P^j_{S}(t-1) \frac{1}{\theta_A^{ij}(t)} \prod_{l \in \partial j} \frac{\theta_A^{ij}(t)\theta_B^{ij}(t)}{\theta_A^{ij}(t-1)\theta_B^{ij}(t-1)} 1[t \neq 0]
\]

\[
- \sum_{a \in \partial j \setminus i} \lambda_{ja}^A(t+1)P^j_{S}(t) \left[ 1 - \frac{p\phi_A^{ij}(\theta_A^{ij}(t))}{\theta_A^{ij}(t)} \prod_{l \in \partial j \setminus a,i} \left( 1 - \frac{\phi_A^{ij}(\theta_A^{ij}(t))}{\theta_A^{ij}(t)} \right) \right] \prod_{l \in \partial j \setminus a} \left( 1 - \frac{\phi_B^{ij}(\theta_B^{ij}(t))}{\theta_B^{ij}(t)} \right) 1[t \neq T]
\]

\[
- \sum_{a \in \partial j \setminus i} \lambda_{ja}^B(t+1)P^j_{S}(t) \left[ \frac{\phi_A^{ij}(\theta_A^{ij}(t))}{\theta_A^{ij}(t)} \prod_{l \in \partial j \setminus a,i} \left( 1 - \frac{\phi_A^{ij}(\theta_A^{ij}(t))}{\theta_A^{ij}(t)} \right) \right] \prod_{l \in \partial j \setminus a} \left( 1 - \frac{\phi_B^{ij}(\theta_B^{ij}(t))}{\theta_B^{ij}(t)} \right) 1[t \neq T]
\]

\[
- \lambda_{j}^S(t+1)P^j_{S}(t) \left[ \prod_{l \in \partial j} \left( 1 - \frac{\phi_B^{ij}(\theta_B^{ij}(t))}{\theta_B^{ij}(t)} \right) \right] \left( 1 + \frac{1}{\theta_A^{ij}(t)} \prod_{l \in \partial j \setminus a} \left( 1 - \frac{\phi_A^{ij}(\theta_A^{ij}(t))}{\theta_A^{ij}(t)} \right) \right) 1[t \neq T]
\]

\[
- \lambda_{j}^B(t+1)P^j_{S}(t) \left[ \prod_{l \in \partial j \setminus i} \left( 1 - \frac{\phi_A^{ij}(\theta_A^{ij}(t))}{\theta_A^{ij}(t)} \right) \right] \left( 1 - \prod_{l \in \partial j \setminus a} \left( 1 - \frac{\phi_B^{ij}(\theta_B^{ij}(t))}{\theta_B^{ij}(t)} \right) \right) 1[t \neq T]
\]

\[
\frac{\partial L}{\partial \nu_i(0)} = -\lambda_i^B(0) - \lambda_i^A(0) - \sum_j \lambda_{ij}^A(0) - \sum_j \lambda_{ij}^A(0) - \sum_j \lambda_{ij}^A(0) - \lambda_{ij}^A(0)
\]

\[
+ \lambda_{ij}^S(0) + \lambda_{ij}^S(0) + \sum_j \lambda_{ij}^S(0) + \sum_j \lambda_{ij}^S(0) + \epsilon \left( \frac{1}{\nu(0)} - \frac{1}{1 - \nu(0)} \right) = 0
\]

(D6)

(D7)
The equations for $\frac{\partial L}{\partial P^i_{B \rightarrow j}}(t), \frac{\partial L}{\partial P^j_{B \rightarrow i}}(t), \frac{\partial L}{\partial \hat{P}^i_{B \rightarrow j}}(t), \frac{\partial L}{\partial \hat{P}^j_{B \rightarrow i}}(t)$ are similar to their $A$ process counterparts with an exchange of variables $A \leftrightarrow B$.

For simplicity one can write equation (D7) as:

$$\frac{\partial L}{\partial \nu_i}(0) = -\lambda B u(0) + \psi_i + \epsilon \left( \frac{1}{\nu_i(0)} - \frac{1}{1 - \nu_i(0)} \right) = 0 \quad (D8)$$

The same method as in [33] is used here to solve the quadratic equation (D8), writing $\nu_i(0)$ as a function of $\psi_i$:

$$\nu_i(0) = \frac{\lambda B u(0) + \psi_i - 2 \epsilon \pm \sqrt{(-\lambda B u(0) + \psi_i)^2 + 4 \epsilon^2}}{-2 \lambda B u(0) + 2 \psi_i} \quad (D9)$$

In this scenario, the positive square root solution satisfies restriction (D2). Given the budget restriction one can obtain $\nu_i(0)$ numerically through

$$\sum_i \nu_i(0) = B_v \quad (D10)$$

One can use the following update procedure to obtain optimal solution iteratively:

**Step 1:** Carry out the forward iteration using given initial conditions, Eqs.(A4)-(A9)

**Step 2:** Compute all Lagrangian multipliers at $t=T$.

**Step 3:** Compute $\lambda^S_{ij}(t-1), \lambda^S_i(t-1), \lambda^A_i(t-1), \lambda^B_i(t-1), \lambda^\hat{A}_{ij}(t-1), \lambda^\hat{B}_i(t-1), \lambda^S_i(t-1)$ using obtained Lagrangian multipliers at time $t$ from Eqs.(D5)-(D6).

**Step 4:** Compute $\lambda^A_{ij}(t-1), \lambda^B_{ij}(t-1), \lambda^\hat{A}_i(t-1), \lambda^\hat{B}_i(t-1), \lambda^S_i(t-1)$ using obtained Lagrangian multipliers and Eqs.(D5)-(D6).

**Step 5:** Compute $\lambda^\hat{A}_{ij}(t-1), \lambda^\hat{B}_i(t-1), \lambda^S_i(t-1)$ using obtained Lagrangian multipliers and Eqs.(D5)-(D6).

**Step 6:** Compute $\lambda^\theta_A_{ij}(t-1), \lambda^\theta_B_{ij}(t-1)$ using obtained Lagrangian multipliers.

**Step 7:** Repeat Steps 1-5 back in time until all Lagrangian multipliers have been obtained for the range $t=0, \ldots, T$.

**Step 8:** Solve Eqs. (D9) and (D10) numerically and update $\nu_i$.

**Step 9:** Repeat Step 1 to Step 8 until convergence.

**Step 10:** Calculate the posterior marginals on the basis of the messages using Eqs.(A9).

A similar approach is used in collaborative processes using the corresponding equations for the dynamics and the Lagrange multipliers’ dynamics.
Appendix E: Optimization of Competitive Processes - Example

We also demonstrate the efficacy of DMP-optimal algorithm on the denser network of the 1994 world metal trade network [45]. The network consists of 80 countries; we will ignore their respective trade volumes and use the dense topology for the current example. The infection probabilities used for the two processes are $p_A = p_B = 0.5$; the budget for $B$ is 1, allocated at node 1 (Argentina) at $t=0$, and a budget of 1 per time step is assigned using the DMP-optimal strategy for process $A$ within the time window of $T = 3$. The results are shown in Fig. 10, where the subfigures represent the containment of process $B$ at different times. The heat bar represents the dominating process per node through the value $P_A^i(t) - P_B^i(t)$. Red/blue represent dominating processes $A/B$, respectively.

FIG. 9. The 1994 network of world trade in metal. The size of nodes represents the degree of nodes [46]. Figure courtesy of Hanyu Zhang.
FIG. 10. The 1994 world metal trade network. The infection parameters used for the two processes are $p_A = p_B = 0.5$, the budget for $B$ is 1, allocated at node 1 (Argentina) at $t = 0$. A budget of 1 per time step is deployed using DMP-optimal for process $A$ for a time window of $T = 3$. The subfigures represent the containment of process $B$ at the different times. The heat bar represents the dominating process per node through the value $P_A^i(t) - P_B^i(t)$. Red/blue represent dominating processes $A/B$, respectively.

Appendix F: Optimization of Collaborative Processes

Optimization in collaborative processes is highly relevant in many health-related problems such as the co-epidemic of HIV and tuberculosis, and in the marketing of correlated products or opinions. In the co-epidemic case, identifying the most catastrophic consequences health-risk is an important aspect of evaluating the risk of a co-epidemic event; while in the marketing area, it aims at the best joint advertising campaign.

We consider two main scenarios in our model: (a) The multi-agent seeding problem, i.e., computing the optimal allocation of one agent which minimizes the number of susceptible vertices at a certain time $T$ for a given spread of the second agent. (b) Optimal allocation of vaccines against one of the processes for maximizing the impact on the spread of both processes. Assume we do not have any vaccines for HIV but vaccines for tuberculosis are available (such as Bacillus Calmette-Gérin [47]); the optimization problem would be that given a certain budget of vaccines for tuberculosis, what is the best vaccination policy to minimize the spreading of both.

The Lagrangian used is similar to the one of the competitive case but with different dynamics and initial conditions ($D$ and $I$ in Eq.(16), respectively). The dynamics equations are enforced through a set of Lagrange multipliers $\lambda$ as
before

\[
D = \sum_{ij} \sum_{t=0}^{T-1} \lambda_{ij}^{DA}(t+1) \left[ \theta_{A}^{i+j}(t+1) - \theta_{A}^{i+j}(t) + P_A \phi_{A}^{i+j}(t) \right] + \sum_{ij} \sum_{t=0}^{T-1} \lambda_{ij}^{DB}(t+1) \left[ \theta_{B}^{i+j}(t+1) - \theta_{B}^{i+j}(t) + P_B \phi_{B}^{i+j}(t) \right]
\]

\[
+ \sum_{ij} \sum_{t=0}^{T-1} \lambda_{ij}^{S}(t+1) \left[ \phi_{A}^{i+j}(t+1) - \phi_{A}^{i+j}(t) + P_A \phi_{A}^{i+j}(t) - P_{A}^{i+j}(t+1) + P_{A}^{i+j}(t) \right]
\]

\[
+ \sum_{ij} \sum_{t=0}^{T-1} \lambda_{ij}^{G}(t+1) \left[ \phi_{B}^{i+j}(t+1) - \phi_{B}^{i+j}(t) + P_B \phi_{B}^{i+j}(t) - P_{B}^{i+j}(t+1) + P_{B}^{i+j}(t) \right]
\]

\[
+ \sum_{ij} \sum_{t=0}^{T-1} \lambda_{ij}^{A}(t+1) \left[ P_{A}^{i+j}(t+1) - P_{A}^{i+j}(t) - P_{S}^{i+j}(t) \left[ 1 - \prod_{k \in \partial \Delta_{ij}} \left( 1 - \frac{p_A \phi_{A}^{k-i}(t)}{\theta_{A}^{k-i}(t)} \right) \right] \right]
\]

\[
- P_{B_{ij}}^{i+j}(t) \left[ 1 - \prod_{k \in \partial \Delta_{ij}} \left( 1 - \frac{p_B \phi_{B}^{k-i}(t)}{\theta_{B}^{k-i}(t)} \right) \right] \]

\[
+ \sum_{ij} \sum_{t=0}^{T-1} \lambda_{ij}^{B}(t+1) \left[ P_{AB}^{i+j}(t+1) - P_{A}^{i+j}(t+1) - P_{B}^{i+j}(t+1) - P_{S}^{i+j}(t+1) + 1 \right]
\]

\[
+ \sum_{ij} \sum_{t=0}^{T-1} \lambda_{ij}^{A}(t+1) \left[ P_{A_{ij}}^{i+j}(t+1) - P_{A}^{i+j}(t+1) - P_{i}^{i+j}(t) \right] \]

\[
(\text{F1})
\]
The initial conditions in this case are of the form:

\[
I = \sum_{ij} \lambda_{ij}^A(0) \left( \theta_A^{i\rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^B(0) \left( \theta_B^{i\rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^B(0) \left( \phi_B^{i\rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^B(0) \left( \phi_B^{i\rightarrow j}(0) - 1 \right)
\]

\[
+ \sum_{ij} \lambda_{ij}^A(0) \left( \phi_A^{i\rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^B(0) \left( \phi_B^{i\rightarrow j}(0) - 1 \right) + \sum_{ij} \lambda_{ij}^B(0) \left( \phi_B^{i\rightarrow j}(0) - 1 \right)
\]

Appendix G: Optimization of Vaccine Allocation

The multi-process seeding problem in the collaborative case has a similar structure to that of the containment task in competitive processes. The vaccine allocation problem is of a slightly different nature. There are many ways in which it can be formulated, the model illustrated below is the one we have chosen to use.

If a node receives a unit of vaccine before being exposed to infected neighbors, it will be immune to that process (denoted as \(B\) in our model), which means the infection message from its neighbor will decrease to 0. However, sometime a node receives part of a unit of vaccine; specifically, if one allocate a certain amount of vaccine, say \(b\) to a node, the probability of it being infected will decrease to \(p_B - b\) (bounded by 0 from below), where \(p_B\) is the initial infection parameter. Meanwhile, the parameter \(p_{BA}\) will also decrease to \(p_{BA} - b\). Therefore the budget for vaccines can be expressed by the budget for decreasing \(p_B\) and \(p_{BA}\) in Eq.(B1)-(B5), which now take the following form:

\[
\theta_B^{i\rightarrow j}(t) = (p_B - b(j)) \phi_B^{i\rightarrow j}(t-1)
\]

\[
\phi_B^{i\rightarrow j}(t) = (p_B - b(j)) \phi_B^{i\rightarrow j}(t-1) + \phi_B^{i\rightarrow j}(t) - P_B^i(t) - P_B^j(t-1)
\]

\[
P_B^i(t) = P_B^i(t-1) + P_B^i(t-1) \left( 1 - \prod_{k \in \partial i} \left( 1 - \frac{(p_B - b(i)) \phi_B^{i\rightarrow k}(t-1)}{\theta_B^{i\rightarrow k}(t-1)} \right) \right)
\]

\[
+ P_B^i(t-1) \left( 1 - \prod_{k \in \partial i} \left( 1 - \frac{(p_B - b(i)) \phi_B^{i\rightarrow k}(t-1)}{\theta_B^{i\rightarrow k}(t-1)} \right) \right)
\]

\[
P_B^i(t) = P_B^i(t-1) + P_B^i(t-1) \left( 1 - \prod_{k \in \partial i \setminus j} \left( 1 - \frac{(p_B - b(i)) \phi_B^{i\rightarrow k}(t-1)}{\theta_B^{i\rightarrow k}(t-1)} \right) \right)
\]

\[
+ P_B^i(t-1) \left( 1 - \prod_{k \in \partial i \setminus j} \left( 1 - \frac{(p_B - b(i)) \phi_B^{i\rightarrow k}(t-1)}{\theta_B^{i\rightarrow k}(t-1)} \right) \right)
\]
The budget restriction in this case is:

\[ B = \lambda Bu \sum_i (b(i) - Bu) \quad \text{(G2)} \]

where \( Bu \) is the total vaccination budget. Another restriction for \( b(i) \) allocated on a node (not exceeding the infection probability) is expressed similarly to before as:

\[ P = \epsilon [\log(b(i)) + \log(p_B - b(i))] \quad \text{(G3)} \]

Finally, the derivative of the part of the Lagrangian that enforces the dynamics \( D \) is differentiated with respect to \( b(i) \), required to complete the set of equations provides

\[
\frac{\partial D}{\partial b(i)} = \sum_{t=0}^{T-1} \left[ - \sum_{k \in \partial i} \lambda^B_{ki} (t+1) \phi_{B,i}^{k \rightarrow i}(t) - \sum_{k \in \partial i} \lambda^B_{ki} (t+1) \phi_{B,i}^{k \rightarrow i}(t) \\
+ \sum_j \lambda^B_{ij}(t+1) P_{S}^{i \rightarrow j}(t) \left( \sum_{l \in \partial(i,l)} \frac{\phi^{l \rightarrow i}_{B}(t)}{\theta^{l \rightarrow i}_{B}(t)} \prod_{k \in \partial(i,l),l} \left(1 - \frac{(p_B - b(i)) \phi_{B,i}^{k \rightarrow i}(t)}{\theta_{B,i}^{k \rightarrow i}(t)} \right) \right) \\
+ \sum_j \lambda^B_{ij}(t+1) P_{A_e}^{i \rightarrow j}(t) \left( \sum_{l \in \partial(i,l)} \frac{\phi^{l \rightarrow i}_{B}(t)}{\theta^{l \rightarrow i}_{B}(t)} \prod_{k \in \partial(i,l),l} \left(1 - \frac{(p_BA - b(i)) \phi_{B,i}^{k \rightarrow i}(t)}{\theta_{B,i}^{k \rightarrow i}(t)} \right) \right) \quad \text{(G4)} \right]
\]

The same procedure (following after Eq.(D8)) can be implemented for the optimization in this case.