Unifying Diffusion Models on Networks and Their Influence Maximisation

Yu Tian†  Renaud Lambiotte†

Abstract

Information diffusion in social networks is a central theme in computational social sciences, with important theoretical and practical implications, such as the influence maximisation problem for viral marketing. Two widely adopted diffusion models are the independent cascade model where nodes adopt their behaviour from each neighbour independently, and the linear threshold model where collective effort from the whole neighbourhood is needed to influence a node. However, both models suffer from certain drawbacks, including a binary state space, where nodes are either active or not, and the absence of feedback, as nodes can not be influenced after having been activated. To address these issues, we consider a model with continuous variables that has the additional advantage of unifying the two classic models, as the extended independent cascade model and the extended linear threshold model are recovered by setting appropriate parameters. For the associated influence maximisation problem, the objective function is no longer submodular, a feature that most approximation algorithms are based on but is arguably strict in practice. Hence, we develop a framework, where we formulate the influence maximisation problem as a mixed integer nonlinear programming and adopt derivative-free methods. Furthermore, we propose a customised direct search method specifically for the proposed diffusion model, with local convergence. We also show that the problem can be exactly solved in the case of linear dynamics by selecting nodes according to their Katz centrality. We demonstrate the rich behaviour of the newly proposed diffusion model and the close-to-optimal performance of the customised direct search numerically on both synthetic and real networks.

1 Introduction

The rapid growth of online social networks, such as Facebook and Twitter, allows hundreds of millions of people worldwide to interact with each other, providing access to a vast source of information on an unprecedented scale. The diffusion of information, opinion, innovation, rumour, etc., is a critical component to explain, for example, how a piece of information could quickly become pervasive through a network via “word-of-mouth” propagation [11,33]. Accordingly, understanding how information spreads in social networks is a central theme in computational social sciences, with theoretical and practical implications, such as the adoption of political viewpoints in presidential elections and the influence maximisation problem for viral marketing [9,14,27]. Information diffusion is a vast research domain, attracting expertise from various fields including mathematics, physics, and biology [31,35,36]. Specifically, one can distinguish two main classes of diffusion models, widely adopted in the context of influence maximisation: the independent cascade (IC) model, and the linear threshold (LT) model, where nodes adapt their behaviour from each neighbour independently, or from the collective influence of the whole neighbourhood, respectively [23,37]. It is well known that both models, which we refer to as “classic models”, suffer from limitations. First of all, the state space of both models is discrete and finite, as nodes can only have state values either 0 (inactive) or 1 (active). Moreover, there is no feedback mechanism in both processes, as nodes can only stay active after being activated, thus may not influence back nodes that influenced them. These limits call for more general models allowing to consider dynamics with feedback between states and more heterogeneity in the agents’ behaviour.

†Funding: The first author is funded by the EPSRC Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with Tesco PLC. The second author acknowledges support from the EPSRC Grants EP/V013068/1 and EP/V03474X/1.

†Mathematical Institute, University of Oxford, Oxford, OX2 6GG, United Kingdom (yu.tian@maths.ox.ac.uk, renaud.lambiotte@maths.ox.ac.uk).
In parallel to this line of work, simple and complex contagions have attracted much research interests in mathematical sociology and physics communities [12, 21]. Essentially, complex diffusion considers situations when the reinforcement of a signal favours its future adoption, which can be modelled via deterministic threshold models. However, their focus is usually on understanding the effects of specific structures, the presence of long ties or the density of triangles for example, and they also tend to consider binary state variables. For models with continuous variables, more has been done within the field of opinion dynamics, where linear models build on the heat equation [32], such as in the DeGroot model [17], and non-linear models include the bounded confidence model for example [16]. A first contribution of this work is to introduce a non-linear, deterministic model for information diffusion which relaxes the constraints of binary variables while allowing feedback between states, and also possesses the IC model and the LT model as limiting cases, thus providing a unifying framework for information diffusion.

As a second contribution, we consider the important problem of influence maximisation (IM), known to have potential applications in various domains [27, 28, 38]. Given a network and an associated diffusion process, the classic IM problem consists in selecting a small set of nodes to activate initially with the aim to maximise the influence spread, commonly defined as the number of activated nodes at the end of the process. Kempe et al. [23] formulated it as a stochastic combinatorial optimisation problem under the classic models, and proposed a greedy algorithm with theoretical approximation guarantees. The near-optimal asymptotic bounds of this seminal work has triggered a vast amount of research in this direction, mostly to further reduce the running time [20, 28]. There are also heuristic solutions, for example centrality-based methods and generic algorithms, but without theoretical guarantee of the performance [5].

However, the aforementioned theoretical guarantees are obtained under the assumption that the influence spread is a submodular function, which is arguably strict [29], and even for the LT model, occurs only for specific choice of the thresholds’ distribution. The IM problem when the thresholds are given by other distributions, even deterministic, is relatively unknown. Therefore, to consider a diffusion model as general as the one we propose in the IM problem, we have developed a brand new framework, where we formulate the IM problem as a mixed integer nonlinear programming (MINLP), with both continuous and integer variables while the objective function can be nonlinear. The exact methods for MINLP are mostly based on branch-and-bound plus convex programming, and require first-order information [6, 8, 10], which is not generally available in the IM problem. Therefore, we treat the objective function as a black box and consider derivative-free methods [8], with the mesh adaptive direct search method as a general solution [1]. Furthermore, we propose a direct search method with local convergence, specifically designed for the general diffusion model.

The primary goal of our paper is to address the drawbacks of the two classic information diffusion models, the IC model and the LT model, while link the research interests in simple and complex contagions. This is achieved by extending both models for continuous state variables while allowing feedback between states in the deterministic case. We then go one step further to propose a general class of information diffusion model interpolating the two extended classic models, where it has the widely accepted features in both models, and can also be freely reduced to either classic model by setting appropriate model parameters. It is the first model of this type, to the best of our knowledge. A second contribution of our paper is a general framework for the IM problem which is not restricted to submodular objective functions, consisting of a MINLP formulation and the derivative-free methods as general solutions. Furthermore, we propose the customised direct search method based on the features associated with the general diffusion model, which can also be applied to classic IM problem of combinatorial optimisation, with local convergence.

The rest of our paper is organized as follows. In section 2 we discuss in detail the IC model and the LT model, and also review the related work in MINLP and derivative-free methods. In section 3 we propose the general class of information diffusion model, where we show its connection and differences with the extended IC model in section 3.2 and the extended LT model in section 3.3. In section 4 we propose the general framework for the IM problem, where we formulate the problem as a MINLP in section 4.1, give the general solutions in section 4.2, analyse the increasingly special cases in section 4.3, and propose the customised direct search in section 4.4. We experimentally verify the features of the general diffusion model, and evaluate the performance of the customised direct search method on both synthetic and real networks in section 5. Finally, we conclude with some implications for future work in section 6. In the appendices, we include the theoretical results of stochastic block models, further features of the diffusion model, specifically
the coexistence of both types of dynamics and the derivative information, and the experiments verifying the time complexity of the customised direct search method.

2 Related work

Here, we first describe the two classic information diffusion models, the independent cascade (IC) model and the linear threshold (LT) model, in more detail [23, 37]. In both models, each individual node has two state values, either 0 for being inactive or 1 for being active, and the primary focus is on progressive processes, where nodes can switch from being inactive to being active but not the other way round. Importantly, there is no feedback between nodes, even on undirected networks, as a node can not be influenced by a node that it influenced before. Specifically, in the LT model, a node \( v_j \) is influenced by each neighbour \( v_i \) according to a weight \( b_{ij} \) s.t. \( \sum_i b_{ij} \leq 1 \), where \( b_{ij} = 0 \) if node \( v_i \) is not \( v_j \)'s neighbour; each node \( v_j \) chooses a threshold \( \theta_j \) uniformly at random from the range \([0, 1]\), which represents the critical influence weight necessary for node \( v_j \) to be activated. Given a random choice of the thresholds, and an initial set of active nodes \( A_0 \), the diffusion process unfolds deterministically in discrete time steps, where in step \( t > 0 \), all nodes that are active in step \( t - 1 \) remain active, and an inactive node \( v_j \) will be activated if the total weight of its active neighbours is at least its threshold \( \theta_j \), i.e.

\[
\sum_{v_i \in A_{t-1}} b_{ij} \geq \theta_j,
\]

where set \( A_{t-1} \) contains the active nodes in step \( t - 1 \). However, in the IC model, each activated node \( v_i \) has a single chance to activate each currently inactive neighbour \( v_j \) when it first becomes active, with success probability \( p_{ij} \) independently of the history thus far. If \( v_j \) is successfully activated, it will have value 1 in the next time step, but whether or not \( i \) succeeds, it cannot further attempt to activate its neighbours in the subsequent rounds. Kempe at al. [23] also considered the variants of both models with feedback, by constructing a multilayer network, with one layer for each time step, and applying the no-feedback models to it. However, they required a predetermined time limit, i.e. the depth of the diffusion.

The IM problem under the two classic diffusion models is NP hard, and the key algorithmic breakthrough lies in the approximation guarantees for the greedy hill-climbing algorithms [23]. Subsequently, several methods have been proposed to further improve the efficiency of the greedy algorithms, maintaining the same approximation guarantees [20, 28], or not exactly [7, 15, 42]. However, one vital assumption to attain the theoretical approximation factor of \((1 - 1/e)\) is that the information spread is submodular, which is not general enough in certain scenarios [29]. Specifically, a function \( f : P(U) \rightarrow \mathbb{R}^+ \cup \{0\} \), where \( P(U) \) is the power set of a finite set \( U \), is submodular, if the marginal gain from adding an element to a set \( S \) is at least of the same level as that from adding the same element to a superset of \( S \), i.e.

\[
f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T),
\]

for all element \( v \in U \) and \( S \subseteq T \subseteq U \). For the LT model, the key correspondence lies in the uniform distribution of thresholds, and it is straightforward to show that the influence spreading under the LT model with deterministic thresholds is not submodular. Furthermore, with deterministic thresholds, it is NP-hard to approximate the IM problem to within a factor of \( n^{1-\epsilon} \) for any \( \epsilon > 0 \) [13, 24]. Therefore, it remains an open problem how to efficiently solve the IM problem under the deterministic LT model.

Mixed integer nonlinear programming (MINLP) problems combine the combinatorial difficulty of optimising over discrete variables with the challenges of handling nonlinear functions, with a wide variety of applications [8]. The global optimisation algorithms of MINLP are mostly based on a branch-and-bound framework and techniques in convex programming, with various attempts to simplify the problem, including preprocessing, convex underestimators, cutting planes, branching, and bounding. There are also heuristic methods without convergence guarantee, tabu search and generic algorithms for instance. We refer the readers to the work of Burer and Letchford [10] and Belotti et al. [6] for comprehensive reviews on the algorithms. In almost all above methods, first-order information of the objective function is required, which is not necessarily available in a MINLP problem.
Derivative-free methods (DFMs), on the other hand, only need function evaluations, and are particularly useful when the underlying function is black-box, or the first-order information is unavailable, prohibitively expensive or deceptive. Recent advances in DFMs either rely on simply input-output data, which yields the direct search methods, or apply an intermediate step to fit a cheaper and smooth approximate function, which results in the model-based methods. We refer the readers to the work of Boukouvala et al. [8] for a thorough review of DFMs in conjunction with MINLP problems. A common assumption in global algorithms is that the objective function is Lipschitz continuous, which is not necessarily true, while in the discontinuous case, the only theoretical result is in direct search methods, of local-convergence-like properties [10]. Hence, DFMs with discontinuous functions, and generally the derivative-free optimisation, remain challenging open problems.

3 General class of information diffusion model

In this section, we propose a novel class of information diffusion model, with continuous state variables while allowing the feedback between states, aiming at circumventing limitations of the classic models, e.g., the IC model and the LT model, in their deterministic case. The full description of the proposed model is in section 3.1. More importantly, the proposed model interpolates between the two classic models after appropriate extensions. We show that it can be reduced to the extended IC model in section 3.2.1 and the extended LT model in section 3.3.1 as well as the corresponding differences in sections 3.2.2 and 3.3.2 respectively.

3.1 Model description

Consider a social network $G(V,E)$, where $V = \{v_1, v_2, \ldots, v_N\}$ is the node set, and $E = \{(v_i, v_j) : \text{there is an edge from node } v_i \text{ to node } v_j\}$ is the edge set. Each edge $(v_i, v_j)$ is considered as a channel connecting nodes $v_i$ and $v_j$ from which the information flows, and can be associated with a weight $W_{ij}$, for example, indicating the level of trust. The value $x_i(t) \in \mathbb{R}$ describes the state, or the amount of information accepted, of node $v_i$ at each time step $t \geq 0$, with the initial state $x_i(0)$ given, and $x(t) = (x_i(t))$ denotes the vector consisting of $x_i(t)$.

We construct our model as follows. Suppose that at time step $t > 0$, an amount $y_j(t; i)$ information is sent to node $v_j$ from its neighbour $v_i$ in order to influence it. It is reasonable to assume that this value is proportional to the accepted information of $v_i$ at previous step, $x_i(t-1)$. We assume further that this relation between $v_j$ and $v_i$ is independent of the following quantities: (i) $x_i(t')$, $\forall t' < t-1, v_i$, so that only the information at the previous time step impacts the present state, in a Markov-like fashion, (ii) $x_j(t-1)$, so that each node receives information from its neighbours independently of its own state, and (iii) $x_l(t-1)$, $\forall l \neq i,j$, which aligns with the notion of independent attempts from the IC model. Moreover, the information flows from node $v_i$ to node $v_j$ through the edge connecting them, and we assume the edge weight $W_{ij}$ contains all the factors that many affect, either enlarge or reduce, the transmission of information. Therefore explicitly, we assume $y_j(t; i) = W_{ij}x_i(t-1)$. However, the decision to accept the overall information received, $y_j(t) = \sum_i y_j(t; i)$, is based on the collective behaviour of the whole neighbourhood. This step is implemented by applying a non-linear transformation to $y_j(t)$, in order to capture how the accumulated information received from all the neighbours transforms into a change for the state of $v_j$, which is reminiscent of the mechanisms of the LT model, but also of non-linear models for opinion dynamics [39]. Specifically, we assume that there is a lower bound $b_t$ in the model, corresponding to the critical mass to trigger the diffusion, s.t. $x_j(t) = 0$ if $y_j(t) < b_t$; symmetrically, there is also an upper bound $B_t$, for the saturation effect at each step [2][18], s.t. $x_j(t) = B_t$ if $y_j(t) > B_t$. Therefore, the general class of information diffusion model we propose is a bounded-linear dynamics.

$$x(t+1) = f_{t+1}(W^T x(t)), \quad \forall t \geq 0,$$

where,

$$f_t(x) = \begin{cases} 
0, & x < b_t, \\
 x, & b_t \leq x < B_t, \\
 B_t, & x \geq B_t,
\end{cases}$$

$\mathbf{W}$
is the time-dependent bound function and is applied element-wise, \( W = (W_{ij}) \) with \( W_{ij} \geq 1 \) is the (weighted) adjacency matrix of the underlying network, \( \{b_t \geq 0\} \) and \( \{B_t \geq 0\} \) are the time-dependent lower and upper bounds, respectively. The initial states \( x(0) \) are given, with \( 0 < b_0 \leq B_0 \). In the general diffusion model, (2) determines the time evolution of influence at time \( t \). In the following, we use the aggregated state value, \( \sum_i x_i(t) \), to represent the overall influence on each node \( v_i \), and a node \( v_i \) is influenced, or active, at time \( t \) if \( x_i(t) > 0 \). The following condition is required to guarantee the convergence of the aggregated state value of each node,

\[
\sum_{t=1}^{\infty} B_t < \infty.
\] (3)

In order to interpret the general diffusion model \( (2) \) and the underlying process more intuitively, we construct a small undirected social network with six agents, as in figure 1, and assign an uniform weight \( W_{ij} = 0.4 \), \( \forall (v_i, v_j) \in E \). For illustrative purpose, we apply the bounds \( b_t = 0.8^t \) and \( B_t = 2 \times 0.8^t \), \( \forall t \geq 0 \), so that condition (3) holds. At \( t = 0 \), suppose node \( v_1 \) is activated with state value 2, thus \( x(0) = [2, 0, 0, 0, 0, 0]^T \). In this configuration, \( v_1 \) can successfully transmit information to, or influence, all its neighbours at \( t = 1 \), i.e. \( x_1(1) > 0 \) if node \( v_j \) is \( v_1 \)'s neighbour; but at \( t = 2 \), nodes \( v_2 \) and \( v_4 \) can collectively influence node \( v_3 \) while node \( v_5 \) cannot influence node \( v_6 \) on its own; see figure 1. This implies that the underlying diffusion process can have features close to the IC model as in step 1, where \( v_1 \)'s neighbours can be influenced from the single source \( v_1 \), and also the collective influence requirements towards the LT model as in step 2. Moreover, there is a positive feedback among the nodes \( v_1, v_2, v_3, v_4 \), as they reinforce their states in this configuration.

### 3.2 Extreme I: the linear dynamics

The model reduces to a linear dynamics for the unbounded state variables, \( y(t) = (y_i(t)) \in \mathbb{R}^N \), with the following updating function,

\[
y(t + 1) = W^T y(t), \quad \forall t \geq 0,
\] (4)

where \( W \) is again the (weighted) adjacency matrix of the network, the same as in (2), but the following condition is required to guarantee the convergence of the aggregated state value,

\[
\rho(W) < 1.
\] (5)

The linear dynamics is equivalently the classic IC model extended for continuous state variables in the deterministic case while allowing feedback between states. In the classic probabilistic IC model, each edge

---

Note that in this article, we exclusively consider normal networks.
weight $W_{ij}$ corresponds to the probability that node $v_i$ can influence node $v_j$ in a Bernoulli trial, thus the expected amount of contribution to the state value of node $v_j$ from node $v_i$ in one step. Then for the expected or deterministic IC model, it is natural to (i) extend the state variables to take continuous values, and (ii) multiply the state value of node $v_i$ in the previous step by $W_{ij}$, i.e. $W_{ij}x_i(t - 1)$, as the amount of information accepted by node $v_j$ from node $v_i$ in the current step $t > 0$, independently of the others, for each pair of nodes $v_i, v_j$. Furthermore, in line with allowing feedback between states, we also relax the requirement that each newly activated node only has a single change to activate its neighbours. We instead assume no-memory for the state values: the ability to either receive or send information at current step is independent of its previous state values. Therefore, $x_j(t) = \sum_i W_{ij}x_i(t - 1)$, and the extended IC model has the same updating function as the linear dynamics in (4).

3.2.1 Relation between the models

We show that the general diffusion model in (2) can be reduced to the linear dynamics as in (4) by setting appropriate bounds \{\textit{b}_t\} and \{\textit{B}_t\}.

**Lemma 3.1.** If $b_t \leq b_0w^t \leq B_t$, \forall $t > 0$, where $w = \min\{W_{ij} : W_{ij} > 0\}$ and $w < 1$, in the general diffusion model (3), then there is no threshold effect from the lower bounds, i.e. \forall $t > 0$ and \forall $i$ s.t. $\sum_j W_{ji}x_j(t - 1) > 0$,

$$\sum_j W_{ji}x_j(t - 1) \geq b_t.$$ \hspace{1cm} (6)

**Proof.** We first show that condition \{3\} can hold with $B_t \geq b_0w^t$, since $\sum_t b_0w^t = b_0/(1 - w) < \infty$

We then note that when a node $v_i$ has $\sum_j W_{ji}x_j(t - 1) > 0$, $\exists j$ s.t. $x_j(t - 1) > 0$ and $W_{ji} > 0$, then

$$\sum_j W_{ji}x_j(t - 1) \geq wx_*(t - 1),$$

where $x_*(t - 1) = \min\{x_j(t - 1) : x_j(t - 1) > 0\}$. Hence, we can show that statement \{6\} is true by proving

$$wx_*(t - 1) \geq b_0w^t,$$

by induction. (i) When $t = 1$, $x_*(0) \geq b_0$, thus $wx_*(0) \geq b_0w$. (ii) Suppose that $wx_*(t - 1) \geq b_0w^t$ is true \forall $t \leq t'$. Then when $t = t' + 1$, by the general diffusion model (2), for each node $v_i$ with $x_i(t') > 0$,

$$x_i(t') = f_{t'}(\sum_j W_{ji}x_j(t' - 1)) \geq f_{t'}(wx_*(t' - 1)) \geq f_{t'}(b_0w^t) = b_0w^{t'},$$

where the first inequality is obtained by $f_{t'}(\cdot)$ being a non-decreasing function, the second one is obtained together with the induction hypothesis, and the equality at the end is because $b_t \leq b_0w^t \leq B_t$, \forall $t > 0$. Therefore, $wx_*(t') \geq b_0w^{t'} = b_0w^{t' + 1}$.

**Theorem 3.2.** If $b_t \leq b_0w^t$ and $B_t \geq B_0 \max_j 1^T W_{i,j}^t$, \forall $t > 0$, where $w = \min\{W_{ij} : W_{ij} > 0\}$, \textbf{1} is the all-one vector and $W_{i,j}^t$ is the $j$-th column of matrix $W^t$, while condition \{3\} holds true, the general diffusion model in (3) is equivalent to the linear dynamics in (4).

**Proof.** We first show that $b_t \leq b_0w^t \leq B_t$, \forall $t > 0$. Since each $(i, j)$ element of $W^t$, $W_{i,j}^t$, is equivalently the sum of weights of length-$t$ paths from node $v_i$ to node $v_j$, its smallest nonzero value is no less than that from an artificial path going through the edge of the minimum weight $t$ times, i.e. $w^t$. Hence,

$$\max_j 1^T W_{i,j}^t \geq \max_{i,j} W_{i,j}^t \geq \min\{W_{i,j}^t : W_{i,j}^t > 0\} \geq w^t,$$

where the first inequality is from $W_{i,j}^t \geq 0$, \forall $i, j$, and the second is from the assumption that $w$ exists. Then it is straightforward to show $b_t \leq b_0w^t \leq B_0 \max_j 1^T W_{i,j}^t \leq B_t$ together with $b_0 \leq B_0$. 


We then show that condition (3) can hold with the current choices, by proving the convergence of $B_t = B_0 \max_j W_{t,j}$. Since in the linear dynamics, condition (5) is required, then

$$1^T \left( \sum_{t=1}^{\infty} W^t \right) 1 = 1^T \left( (I - W)^{-1} - I \right) 1 < \infty.$$ 

Hence,

$$\sum_{t=1}^{\infty} \max_j 1^T W_{t,j}^t \leq \left( \sum_{t=1}^{\infty} 1^T W^t \right) 1 < \infty,$$

where the first inequality is from $W_{ij} \geq 0$, $\forall i, j$. Then $\sum_{t=1}^{\infty} B_t^* = B_0 \sum_{t=1}^{\infty} \max_j 1^T W_{t,j}^t < \infty$ follows naturally. Therefore, there are valid bounds with $b_t \leq b_0 w^t$ and $B_t \geq B_0 \max_j 1^T W_{t,j}^t$, $\forall t > 0$, for the general diffusion model in (2).

Finally, we show that given such bounds in (2),

$$x(t)^T = x(0)^T w^t, \quad \forall t > 0,$$

by induction. We note that condition (5) necessarily requires $w < 1$. Then by lemma 3.1, there is no threshold effect from the current lower bounds, thus we will only check the saturation effect at the upper bounds in the following. (i) When $t = 1$, $\max_j x(0)^T W_{t,j} \leq B_0 \max_j 1^T W_{t,j} = B_t$, since $x_1(0) \leq B_0$ and $W_{ij} \geq 0, \forall i, j$. Hence, $x(1)^T = f_1(x(0)^T w) = x(0)^T w$. (ii) Suppose $x(t)^T = x(0)^T w^t, \forall t \leq t'$. Then,

$$\max_j x(t')^T W_{t,j} = \max_j x(0)^T W_{t'}^T \leq B_0 \max_j 1^T W_{t'}^T = B_{t'} + 1,$$

where the first equality is by induction hypothesis, and the inequality is again by $x_1(0) \leq B_0$ and $W_{ij} \geq 0, \forall i, j$. Hence,

$$x(t' + 1)^T = f_{t'+1}(x(t')^T w) = x(t')^T w = x(0)^T w^{t'} + 1.$$ 

It is then straightforward to show that dynamics characterised by (7) has updating function (4). Therefore, the general diffusion model with the specified choice of bounds is equivalent to the linear dynamics.  

### 3.2.2 Differences between the models

We then analyse the deviation from the linear dynamics when changing the bounds, specifically the imposed threshold effect when increasing the lower bounds. We denote the highest possible value of the lower bounds guaranteed to maintain the linear dynamics by $b'_1 = b_0 w^t$, where $w = \min\{W_{ij} : W_{ij} > 0\}$. Once $b_t > b'_1$, there could be dramatic change in the diffusion profiles, where nodes with more overlap in their neighbourhoods of various distances will be able to achieve a higher aggregated state value. Claim 3.1 demonstrates it through a two-block planted stochastic block model (SBM).

**Claim 3.1.** With $b_0 = B_0$, and $\{B_t\}$ as in Theorem 3.2 in the general diffusion model (2), $SBM(p_{in}, p_{out})$ with two equally sized communities, $B_1, B_2$, and uniform weight $\alpha$ has the following properties:

1. when $b_1 \leq b'_1 = b_0 \alpha$, the expected aggregated state value, $E[\sum_i x_i(1)]$, from the initially activated node set $(i) \ A_0 = \{v_{i1}, v_{i2}\} \subset B_1$, is the same as that from (ii) $A_0 = \{v_{j1}, v_{j2} : v_{j1} \in B_1, v_{j2} \in B_2\}$;

2. while when $b'_1 < b_1 \leq 2b'_1$ if $p_{in} \neq p_{out}$,

$$E[\sum_i x_i(1)] \text{ from set (i) is larger than that from set (ii).}$$

---

*In the specific case here, the upper bound $2b'_1$ is equivalent to require that at most two initially activated neighbours are needed to activate a node.*
Proof. For the SBM, $\mathbf{W} = \alpha \mathbf{A}$, where $\mathbf{A}$ is the (unweighted) adjacency matrix. Therefore, for each pair of nodes $v_i, v_j$, $A_{ij} \sim \text{Bernoulli}(p_{ij})$, with

$$p_{ij} = p_{in} \delta(\sigma_i, \sigma_j) + p_{out}(1 - \delta(\sigma_i, \sigma_j)),$$

where $\sigma_i \in \{1, 2\}$ indicates the block membership of each node $v_i$. Further, we denote the linear part of the state vector by $y(t + 1) = \mathbf{W}^T \mathbf{x}(t)$, then for each node $v_i$ at each time $t > 0$,

$$x_i(t) = \mathbf{f}_i(y_i(t)),$$

$$y_i(t) = \sum_j W_{ji} x_j(t - 1) = \alpha \sum_j A_{ji} x_j(t - 1).$$

Then at $t = 1$,

$$y_i(1) = \alpha \sum_j A_{ji} x_j(0) = \alpha b_0 \sum_{v_j \in \mathcal{A}_0} A_{ji}$$

$$= \alpha b_0 \left( \sum_{v_j \in \mathcal{A}_0 \cap \mathcal{B}_{\sigma_i}} \text{Bernoulli}(p_{in}) + \sum_{v_j \in \mathcal{A}_0 \setminus \mathcal{B}_{\sigma_i}} \text{Bernoulli}(p_{out}) \right)$$

$$= \alpha b_0 \left( \text{Bin}(K_{\sigma_i}, p_{in}) + \text{Bin}(K - K_{\sigma_i}, p_{out}) \right),$$

where $\mathcal{A}_0 = \{v_i : x_i(0) > 0\}$ is the (given) set of initially activated nodes, $K = |\mathcal{A}_0|$, and $K_t = |\mathcal{A}_0 \cap \mathcal{B}_t|$, $t = 1, 2$.

Hence, in case (i),

$$y_i(1) = \alpha b_0 \left( \text{Bin}(2, p_{in}) \delta(\sigma_i, 1) + \text{Bin}(2, p_{out}) \delta(\sigma_i, 2) \right),$$

while in case (ii),

$$y_i(1) = \alpha b_0 \left( \text{Bin}(1, p_{in}) + \text{Bin}(1, p_{out}) \right).$$

When $b_1 \leq b_1^*$, $x_i(1) = y_i(1), \forall i$. Then, in case (i),

$$\mathbb{E} \left[ \sum_{i} x_i(1) \right] = \mathbb{E} \left[ \sum_{i} y_i(1) \right] = N_b \times \alpha b_0 \times (2p_{in} + 2p_{out}),$$

and in case (ii),

$$\mathbb{E} \left[ \sum_{i} x_i(1) \right] = \mathbb{E} \left[ \sum_{i} y_i(1) \right] = 2N_b \times \alpha b_0 \times (p_{in} + p_{out}),$$

where $N_b$ is the size of each community. Hence, the expected aggregated state values are the same in the two cases.

However, when $\alpha b_0 = b_1^* < b_1 \leq 2b_1^* = 2\alpha b_0$, in case (i),

$$P(y_i(1) \geq b_1) = P(\text{Bin}(2, p_{in}) \delta(\sigma_i, 1) + \text{Bin}(2, p_{out}) \delta(\sigma_i, 2) = 2) = p_{in}^2 \delta(\sigma_i, 1) + p_{out}^2 \delta(\sigma_i, 2),$$

thus,

$$\mathbb{E} \left[ \sum_{i} x_i(1) \right] = \sum_{i} 0P(y_i(1) < b_1) + 2\alpha b_0 P(y_i(1) = 2\alpha b_0) = N_b \times 2\alpha b_0 \times (p_{in}^2 + p_{out}^2).$$

(9)

While, in case (ii),

$$P(y_i(1) \geq b_1) = P(\text{Bin}(1, p_{in}) + \text{Bin}(1, p_{out}) = 2) = p_{in}p_{out},$$

thus,

$$\mathbb{E} \left[ \sum_{i} x_i(1) \right] = \sum_{i} 0P(y_i(1) < b_1) + 2\alpha b_0 P(y_i(1) = 2\alpha b_0) = 2N_b \times 2\alpha b_0 \times p_{in}p_{out}. \quad (10)$$

Hence, the expectation from (i) as in (9) is larger than the one from (ii) as in (10) by condition (8).
3.3 Extreme II: the LT model

The linear threshold (LT) model that we refer to is the classic LT model extended for continuous variables and deterministic thresholds while allowing feedback between states. The linear-activation strategy is maintained, where a node $v_i$ will be activated at future step $t+1$ if the sum of its neighbours’ current state values, weighted by the edge weights, is not smaller than its threshold $\theta_i$, i.e. $\sum_j W_{ji}x_j(t) \geq \theta_i$.

With continuous variables, we can control the source of nonlinearity to be only this threshold effect, and set the activated state value of each node $v_i$ to be the threshold value $\theta_i$. The state values are then not always 0 or 1 as in the classic LT model, but can change magnitude along time. Hence, it is necessary to impose time-dependent thresholds $\theta_i(t)$. As in section 3.2, we allow feedback between states by assuming the no-memory properties in the state values. Therefore, for each node $v_i$ at time step $t > 0$,

$$
x_i(t) = \begin{cases} 
\theta_i(t), & \sum_j W_{ji}x_j(t) \geq \theta_i(t), \\
0, & \text{otherwise}
\end{cases}
$$

where the initial states at $t=0$ are given.

3.3.1 Relation between the models

We first show that the general diffusion model (2) can be reduced to the LT model with uniform thresholds $\theta_t$ among the nodes, by setting $b_t = B_t = \theta_t$, $\forall t$. In this case, the bound function in the general diffusion model (2) is $f_t(x) = \theta_t H(x - \theta_t)$ where $H(\cdot)$ is the Heaviside step function, thus only has threshold effect at each step $t$. Therefore, the general diffusion model has the same updating function as the LT model in (11). The only step left for the full equivalence is to show that condition (3) holds true.

There are various approaches to prove the convergence of the upper-bound series in (3). Here corresponding to the threshold effect, we consider the following threshold-type bounds,

$$
b_t = (a_b \alpha)^t b_0,
B_t = (a_B \alpha)^t B_0,
$$

$\forall t \geq 0$, where $\alpha = \sum_{(v_i,v_j) \in E} W_{ij} / |E|$ is the mean weight, and $0 \leq a_b \leq a_B$ are the thresholds for the upper and lower bounds, respectively.

**Theorem 3.3.** With $b_0 = B_0$, if the threshold-type bounds have

$$
a_B \alpha < 1 ,
a_b = a_B ,
$$

the general diffusion model (2) with such bounds is equivalent to the LT model in (11). Further, if the underlying network is unweighted or uniformly weighted with $\alpha > 0$, the general diffusion model is equivalent to the constant threshold (CT) model$^a$, where for each node $v_i$ at each time step $t > 0$,

$$
x_i(t) = \begin{cases} 
h_t(\theta), & \sum_{v_j \in A_{t-1}} A_{ji} \geq \theta, \\
0, & \text{otherwise}
\end{cases}
$$

where $\theta \in \mathbb{Z}^+$ is the constant threshold, $A_{t-1} = \{v_j : x_j(t-1) > 0\}$ is the set of nodes of positive state values at time step $t-1$, $A = (A_{ij}) \in \{0, 1\}^{N \times N}$ is the (unweighted) adjacency matrix, and $h_t(\cdot) > 0$ is a time-dependent function.

*Proof.* We first show that under the condition [13], the threshold-type bounds are valid for the general diffusion model, i.e. condition (3) holds true. It is straightforward by the convergence dependence on the

$^a$The CT model here has been extended in the same way as the LT model for continuous variables while allowing feedback between states. It maintains the key feature that a node will have positive state value only if constant number of its neighbours do in the previous step [5].
common ratio $a_B\alpha$ of geometric series, where when $a_B\alpha < 1$, $\sum_{t=1}^{\infty} B_t = a_B\alpha B_0/(1 - a_B\alpha)$ is finite. It is actually the sufficient and necessary condition since when $a_B\alpha \geq 1$, $\sum_{t=1}^{\infty} B_t \geq \sum_{t=1}^{\infty} B_0$ diverges.

Then by condition (14) and the definition of threshold-type bounds (12), $b_t = B_t$, $\forall t \geq 0$. Hence, the corresponding general diffusion model (2) has the same updating function as the LT model in (11) with $\theta_t^i = b_t = B_t$, $\forall i, t$.

Further, $x_i(t) = (a_B\alpha)^t b_0$ if $x_i(t) > 0$, $\forall i, t$. Then, a node $v_i$ can have positive state value at step $t$ if

$$\sum_j W_{ji} x_j(t-1) \geq b_t = (a_B\alpha)^t b_0,$$

$$\sum_{v_j \in A_{t-1}} W_{ji} (a_B\alpha)^{t-1} b_0 \geq (a_B\alpha)^t b_0,$$

$$\sum_{v_j \in A_{t-1}} W_{ji} \geq a_B\alpha.$$

Thus, when the underlying network is unweighted or has uniform weight $\alpha > 0$, the weight matrix is $W = (W_{ij}) = \alpha A$. The condition (16) is then

$$\sum_{v_j \in A_{t-1}} A_{ji} \geq a_B.$$

Thus, the corresponding general diffusion model has the same updating function as the CT model in (15) with $\theta = a_B = a_B$.

Hence, in the case of unweighted or uniformly weighted networks, the bound thresholds $a_B, a_B$ can play the same role as the threshold in the CT model. Specifically, $a_B = a_B = 1$ corresponds to simple contagion, where a node can be influenced by a single influenced neighbour, and $a_B = a_B > 1$ corresponds to complex contagion, where collective effort from the neighbourhood is required to influence a node. For general cases of weighted networks as in (16), the bound thresholds $a_B, a_B$ together with the mean weight $\alpha$ serve the same role as the threshold in the classic LT model as in (1), i.e. on the weighted sum of influenced neighbours in the previous step. Due to the further correspondence by theorem 3.3, we will consider exclusively the threshold-type bounds (12) hereafter.

### 3.3.2 Differences between the models

The general diffusion model (2) introduces a linear part following the jump at the lower bound $b_t$ through the bound function $f_t(\cdot)$ at each time step $t$, allowing the difference between the lower bound threshold $a_B$ and the upper bound threshold $a_B$ in the threshold-type bounds (12). Here, we analyse the deviation from the LT model when increasing the difference between $a_B$ and $a_B$, with $b_0 = B_0$. As the other extreme, when the bounds are sufficiently far from each other, i.e. the lower bounds are sufficiently small and the upper bounds are sufficiently large, the general diffusion model will be equivalent to the linear dynamics as in theorem 3.2 which has dramatically different properties from those of the LT model.

Specifically, we consider increasing the upper bounds through $a_B$, since manipulating the lower bounds has been analysed in section 3.2.2. We examine the locally tree-like effect, where the diffusion can possibly go through the locally tree-like structure, while it can hardly occur when $a_B = a_B > 1$. Claim 3.2 illustrates the case of uniform weights, which corresponds to the averaged behaviour of weighted systems.

**Claim 3.2.** With $b_0 = B_0$ and the threshold-type bounds (12), suppose the network is unweighted or uniformly weighted with $\alpha > 0$, and at a particular time $t^\prime \geq 0$ in the diffusion process by (2), there is a tree-like structure of the influenced nodes $A_v = \{v_j : x_j(t^\prime) > 0\}$, and some node $v_i$ of zero aggregated state value, s.t.

$$\exists v_j^* \in A_v, \text{ s.t. } A_{j^*i} = 1,$$

where $A = (A_{ij})$ is the (unweighted) adjacency matrix. Then
1. when \( a_B = a_b > 1 \), node \( v_i \) can never have positive state value at \( t = t' + 1 \);

2. when \( a_B > a_b > 1 \), node \( v_i \) can have positive state value at \( t = t' + 1 \) given a sufficiently large \( a_B \).

**Proof.** Since the underlying network is unweighted or uniformly weighted, the weight matrix is \( W = \alpha A \). Then, when \( a_B = a_b > 1 \), \( x_j(t) = (a_b \alpha)^t b_0 \), \( \forall v_j \in A_t \) and \( \forall t \), thus

\[
\sum_j W_{ji} x_j(t') = \alpha (a_b \alpha)^t b_0 < a_b \alpha (a_b \alpha)^t b_0 = b_{t'+1},
\]

thus node \( v_i \) cannot have positive state value at time step \( t' + 1 \).

However, when \( a_B > a_b > 1 \), the highest possible value of node \( j^* \) at time step \( t' \) is \( (a_B \alpha)^t b_0 \), thus node \( v_i \) can have positive state value if

\[
W_{j^*i}(a_B \alpha)^t b_0 = \alpha (a_B \alpha)^t b_0 \geq b_{t'+1} = (a_b \alpha)^{t' + 1} b_0,
\]

i.e. \( a_B' \geq a_b^{t'+1} \). \(\square\)

We can take the planted two-block SBM for example again as in section 3.2.2 but for the cases when increasing \( a_B \) from 2 to 4 while maintaining \( a_b = 2 \). Accordingly, we analyse the performance of the following two sets of four initially activated nodes: (i) \( A_0 = \{v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}\} \subset B_1 \); (ii) \( A_0 = \{v_{j_1}, v_{j_2}, v_{j_3}, v_{j_4} : v_{j_1}, v_{j_2} \in B_1, v_{j_3}, v_{j_4} \in B_2\} \). We can show that the increase in aggregated state value from case (i) is expected to be more than that from case (ii) (see appendix A for details), which is numerically verified in section 5.1.

### 4 Influence maximisation

In the previous section, we have discussed the general class of information diffusion model that we propose and its most salient features. Now, we proceed to a key algorithmic problem associated to information diffusion, the influence maximisation (IM) problem, i.e. to maximise the overall influence on the nodes at the end of the process, usually subject to the constraint of a limited number of initially activated nodes, determined by the budget size. In this section, we will first introduce a new formulation of the IM problem in section 4.1 and give general solutions to this task in section 4.2. In these two sections, we focus on the general features of the IM problem when the objective is not necessarily submodular. Then we turn to the special cases when the dynamics is given by the general diffusion model in section 4.3 and further propose a customised algorithm in section 4.4.

#### 4.1 Problem formulation

As described in section 3.1, we use the aggregated state value, \( \sum_t x_i(t) \), to represent the overall influence on each node \( v_i \), following the general class of information diffusion model in (1). Hence, the objective function of the corresponding IM problem is the sum of aggregated state values over all nodes,

\[
F(x(0)) = \sum_{i=1}^{N} \sum_{t=1}^{\infty} x_i(t) = \sum_{t=1}^{\infty} 1^T x(t),
\]

where \( N \) is the network size, \( 1 \) is the all-one vector, and state vector \( x(t) = (x_i(t)) \) is obtained by the general diffusion model (2) where \( x_i(t) \in \{0\} \cup [b_t, B_t] \). The IM problem is then to maximise \( F(x(0)) \) with respect to the initial state vector \( x(0) \), subject to the constraint of limited budget size,

\[
|\{i : x_i(0) > 0\}| \leq K,
\]

where \( K \in \mathbb{Z}^+ \) is the budget size.
Generally, we formulate the IM problem with objective (17) and constraint (18) as a mixed-integer (nonlinear) programming (MINLP),

$$\max_{x,z} F(x)$$

s.t. \( x_i \leq B_0 z_i, \)

\( x_i \geq b_0 z_i, \)

\( \sum_i z_i \leq K, \)

\( x_i \in \mathbb{R}, z_i \in \{0,1\}, \forall i, \)

where \( 0 < b_0 \leq B_0 \) restrict the initial influence value, \( K \in \mathbb{Z}^+ \) is the budget size, and the objective function \( F(\cdot) \) is the overall influence on the whole network as in (17). The variables in vector \( x \) correspond to the initial states, while the variables in vector \( z \), of the same dimension \( N \), are added to appropriately impose the constraint (18).

The difficulty of the optimisation problem lies in the objective function \( F(x) \). (i) \( F(x) \) is not always smooth and even discontinuous, since \( f_t(x) \) in (2) can be nonsmooth at \( B_t \) and discontinuous at \( b_t \). (ii) A closed-form of \( F(x) \) cannot be obtained generally, except when \( f_t(x) = x, \forall t, \) in (2). (iii) The derivative information is rarely very useful in finding a maximal point, as discussed in appendix B.2. However, we can show that the evaluation of the objective function can be solved in \( O(|E|T_{\epsilon}) \) time, where \( |E| \) is the number of edges and \( T_{\epsilon} \) is the number of time steps required for convergence of the underlying diffusion process with tolerance \( \epsilon \). This is a bonus in the deterministic setting [30]. Hence, it is necessary to treat the objective function as an input-output (black-box) system and resort to derivative-free methods.

**Theorem 4.1.** Given a network \( G(V,E) \) with the weight matrix \( W \) and an initial state \( x(0) \), the problem of exactly computing the function \( F(x(0)) \) in (17) can be solved in \( O(|E|T_{\epsilon}) \) time, where \( T_{\epsilon} \) is the number of time steps required for the convergence of the underlying diffusion process with tolerance \( \epsilon > 0 \).

**Proof.** The time complexity follows from algorithm [1]. In each iteration \( t \), each nonzero element of the weight matrix \( W \) has only one chance to be used to potentially adjust the state value \( x^{(t)} \), and there are overall \( O(|E|) \) such elements. Therefore, the time complexity of each iteration is \( O(|E|) \), and the overall evaluation has time complexity \( O(|E|T_{\epsilon}) \), dependent on the number of steps towards convergence, \( T_{\epsilon} \). \( \square \)

**Algorithm 1** Influence evaluation.

1: Input: A network \( G(V,E) \) with its weight matrix \( W \) where \( W_{ij} > 0 \) if \( (v_i,v_j) \in E \), the initial state \( x(0) \) where \( x_i(0) > 0 \) if \( v_i \in A_0 \), model parameters \( \{b_t\}, \{B_t\} \), and the tolerance \( \epsilon \).
2: Output: The objective value \( F \) as in (17).
3: Set \( t \leftarrow 0, x^{(0)} \leftarrow x(0), \) and \( F \leftarrow 0 \).
4: Mark all the out-neighbours of \( A_0 \) as potentially activated nodes, \( B_0 \leftarrow \bigcup_{v_i \in A_0} N^{out}(v_i) \).
5: while \( |x^{(t)}| > \epsilon \) do
6: \( A_1, B_1 \leftarrow \emptyset, \) and \( x^{(t+1)} \leftarrow 0 \);
7: for each potentially activated node \( v_j \in B_0 \) do
8: \( x_j^{(t+1)} = f_{t+1}(\sum_{i \in A_0} W_{ij} x_i^{(t)}); \)
9: if \( x_j^{(t+1)} > 0 \) then
10: \( A_1 \leftarrow A_1 \cup \{v_j\}; \)
11: \( B_1 \leftarrow B_1 \cup N^{out}(v_j); \)
12: \( F \leftarrow F + x_j^{(t+1)}; \)
13: end if
14: end for
15: \( t \leftarrow t + 1, A_0 \leftarrow A_1, \) and \( B_0 \leftarrow B_1; \)
16: end while
Therefore, we can obtain another upper bound for the linear dynamics. We denote the eigenvector with the largest inner product \( u_{\Omega} = \Omega \) where \( \Omega \) are the maximum values at every steps but are not necessarily achieved by all nodes. We approximate them by the smallest \( \epsilon \) such that satisfies
\[
|B_t 1| = B_t \sqrt{N} < \epsilon,
\]
where \( 1 \) is the all-one vector, and \( N \) is the number of nodes in the network. This value is finite by condition \( \Box \), and provides a better approximation when \( \{B_t\} \) are tighter. (ii) When both the upper and the lower bounds are loose, we can consider the convergence of linear dynamics, where \( x(t) = (W W^T)^t x(0), \forall t \). Hence,
\[
|x(t)|^2 = x(t)^T x(t) = x(0)^T (W W^T)^t x(0).
\]
Since \( WW^T \) is normal, it is unitarily diagonalisable, where suppose \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \),
\[
WW^T = \sum_{i=1}^{N} \lambda_i u_i u_i^T,
\]
\[
u_i^T u_i = \delta(i,j) \quad \text{with} \quad \delta(i,j) = 1 \quad \text{if and only if} \quad i = j \quad (0 \quad \text{otherwise}),
\]
\[
\lambda_1 = \rho(W)^2 < 1, \quad \text{by condition} \quad (5) \quad \text{in the linear dynamics. We denote the eigenvector with the largest inner product} \quad u_i^T x(0) \quad \text{by} \quad u_*.
\]
Then,
\[
|x(t)|^2 = \sum_{i=1}^{N} \lambda_i^2 (u_i^T x(0))^2 \leq N \lambda_1^2 (u_1^T x(0))^2 = N (\rho(W)^t u_1^T x(0))^2.
\]
Therefore, we can obtain another upper bound for \( T_\epsilon \) by the smallest \( t \) such that
\[
\sqrt{N} \rho(W)^t u_1^T x(0) < \epsilon.
\]

### 4.2 General solutions

Since we know the exact process to evaluate the objective function, which does not fall in a simple family, e.g. polynomials, model-based methods are not appropriate in this problem. This effectively limits the choice among algorithm’s classes for its solution to that of direct-search methods. Among such algorithms, the mesh adaptive direct search (MADS) method is the only one that has local convergence analysis when the objective function is not necessarily Lipschitz continuous \([40] \). Therefore, we consider the MADS for mixed variables (MV) \([1] \), as a general solution to our problem, which can be implemented by the software NOMAD \([3, 26] \). Here, we provide a brief overview of MADS, which paves the way for the customised algorithm proposed later in section \( \Box \).

In MADS, each vector \( y = (y^c, y^d) \) is partitioned into its continuous and discrete components, and we denote the maximum dimensions of the continuous and discrete variables by \( n^c \) and \( n^d \) respectively, thus \( y^c \in \Omega^c \subseteq \mathbb{R}^{n^c} \) and \( y^d \in \Omega^d \subseteq \mathbb{Z}^{n^d} \). The general problem \( \Box \) under consideration is given by
\[
\min_{y \in \Omega} f(y),
\]
where \( f : \Omega \rightarrow \mathbb{R} \cup \{\infty\} \), and the domain, or the feasible region, is the union of continuous domain across possible discrete variable values, i.e.
\[
\Omega = \bigcup_{y^d \in \Omega^d} (\Omega^c(y^d) \times \{y^d\}),
\]
where \( \Omega^c(y^d) \) indicates that the continuous domain can change with different discrete variable values, and \( \Omega = \Omega^c \) if \( n^d = 0 \). The constraints are incorporated in the domain, and are treated by the extreme barrier approach \( f_\Omega \), where \( f_\Omega(y) = f(y) \) if \( y \in \Omega \) and \( \infty \) otherwise.

\( \Box \)Note the general MADS method considers the minimisation problem, therefore we transform our objective to be \( \min_{k,a} -F(x) \) when applying the algorithm.
MADS is a local search method, which aims to find a local minimiser, therefore it is important to define the local optimality. For the continuous variables, the neighbourhood is well-defined as the open ball, $B_{\epsilon}(y^p) = \{ y^p \in \mathbb{R}^n : \| y^p - y^p \| < \epsilon \}$ with $\epsilon > 0$. However, different notions of the discrete neighbourhood exist. One common choice for integer variables is $\mathcal{N}(y) = \{ y' \in \Omega : y' = y^c, \| y^d - y^d \|_1 \leq 1 \}$. With a user-defined discrete neighbourhood, the classical definition of local optimality can be extended to mixed variable domains as follows.

**Definition 4.1.** A point $y = (y^c; y^d) \in \Omega$ is said to be a local minimiser of $f$ on $\Omega$ with respect to the set of neighbours $\mathcal{N}(y) \subset \Omega$ if there exists an $\epsilon > 0$ such that $f(y) \leq f(\tilde{y})$ for all $\tilde{y}$ in the set

$$\Omega \cap \left( \bigcup_{y' \in \mathcal{N}(y)} B_{\epsilon}(y^c) \times y^d \right).$$

MADS algorithm is characterised by an optional **search** step, a local **poll** step, and an extended **poll** step, where the objective $f_{\Omega}$ is evaluated at specific points defined on an underlying mesh $M_k$ at each iteration $k$. The goal of each iteration is to find a feasible improved mesh point from current iterate, $y \in M_k$ s.t. $f_{\Omega}(y) < f_{\Omega}(y^{(k)})$, and the algorithm will output the point it converges to. The mesh $M_k$ at each iteration $k$ is a central concept in the method, which is formed as the direct product of $\Omega^c$ with the union of a finite number of lattices in $\Omega^d$,

$$M_k = \bigcup_{l=1}^{l_{\max}} M_k^l \times \Omega^d,$$

where $l = 1, \ldots, l_{\max}$ indicates each combination of discrete variable values, and the lattice $M_k^l$ is defined through previously evaluated points, the positive spanning directions and the mesh size parameter $\Delta_k^m$ dictating its coarseness. In the **poll** step, the method evaluates the discrete neighbourhood $\mathcal{N}(y^{(k)})$, and the points whose continuous part is close to the current iterate along certain directions, $P_k(y^{(k)})$, controlled by $\Delta_k^m$ and the poll size parameter $\Delta_k^d$. The extended **poll** step is triggered when the **poll** step fails to find an improved point. It consists of a finite sequence of **poll** steps performed around the points in $\mathcal{N}(y^{(k)})$ whose objective value is sufficiently close to the incumbent value, i.e. $f_{\Omega}(y^{(k)}) \leq f_{\Omega}(y) \leq f_{\Omega}(y^{(k)}) + \xi_k$, for some user-defined tolerance $\xi_k \geq \xi$ (e.g. $\xi_k = \max\{\xi, 0.05|f(y^{(k)})|\}$), and we denote such set of nodes by $\mathcal{N}_{k}^{\xi_k}(y^{(k)})$. We summarise the main ideas of MADS in algorithm 2.

**Algorithm 2** Mesh adaptive direct search for mixed variables (MADS-MV).

1. **SEARCH** step (optional): Evaluate $f_{\Omega}$ on a finite set of trial points on the mesh $M_k$ (20). If an improved mesh point is found, the **SEARCH** step may terminate, skip the next **POLL** step and go directly to step 4.
2. **POLL** step: Evaluate $f_{\Omega}$ on the set $P_k(y^{(k)}) \cup \mathcal{N}(y^{(k)}) \subset M_k$ (i.e. close to the current iterate), until an improved mesh point is found, or until all points have been exhausted. If an improved mesh point is found, go to step 4.
3. **EXTENDED POLL** step: Perform a finite sequence of **poll** step starting from each point $y \in \mathcal{N}_{k}^{\xi_k}(y^{(k)}) \subset \mathcal{N}(y^{(k)})$ with $f_{\Omega}(y^{(k)}) \leq f_{\Omega}(y) \leq f_{\Omega}(y^{(k)}) + \xi_k$, until an improved mesh point is found or until all points have been exhausted.
4. Parameter update: Coarsen $\Delta_{k+1}^m$ and $\Delta_{k+1}^d$ when an improved mesh point is found and refine them otherwise. Update $\xi_k \geq \xi$, increment $k \leftarrow k + 1$, and go to step 1.

As mentioned before, MADS is among the few algorithms that can relax the assumptions for convergence analysis to include discontinuous functions. To conclude the analysis, we mention that there are many heuristic algorithms in derivative-free methods [25], but without any convergence guarantee.
4.3 Special cases

In the previous section, we analyse the most general features of the IM problem, and give general solutions accordingly. Hereafter, we turn our attention to two special cases of the optimisation problem, in order to provide insights into the general IM problem.

The first case is when the lower bounds, \(\{b_i\}\), are sufficiently small, where we can show that the objective function \(F(\cdot)\) is continuous and concave, with respect to the continuous variables \(x\).

**Theorem 4.2.** If \(\{b_i\}, \{B_i\}\) are as in lemma 3.1 then the objective function \(F(\cdot)\) in (19) is continuous and concave w.r.t. the continuous variables \(x\).

*Proof.* By lemma 3.1 the lower bounds are effectively 0. Therefore, in the general diffusion model (2), \(f_t(x) = 0\) if and only if \(x = 0\), \(\forall t\). Hence, \(f_t(x)\) is equivalent to another bound function \(\tilde{f}_t(x)\) associated with the same upper bound \(\tilde{B}_t = B_t\) but a different lower bound \(\tilde{b}_t = 0\).

We show that \(f_t(x)\) is continuous by the continuity of \(\tilde{f}_t(x)\). It is straightforward since (i) each part of the function \(\tilde{f}_t(x)\), when \(\tilde{b}_t < x < \tilde{B}_t\) or \(x > \tilde{B}_t\), is continuous, and (ii) limits at the boundary points are equal to the function values, \(\lim_{x \to \tilde{b}_t} \tilde{f}_t(x) = \tilde{B}_t = \tilde{f}_t(\tilde{B}_t) = \lim_{x \to \tilde{B}_t^+} \tilde{f}_t(x)\), and \(\lim_{x \to \tilde{B}_t^-} \tilde{f}_t(x) = 0 = \tilde{f}_t(x)\). Hence, the continuity of the objective function \(F(\cdot)\) follows from the property that composite of continuous functions is still continuous.

Then we show that \(f_t(x)\) is concave by the concavity of \(\tilde{f}_t(x)\), i.e. \(\forall x, y\) and \(\alpha \in [0,1]\),

\[
\tilde{f}_t((1-\alpha)x + \alpha y) \geq (1-\alpha)\tilde{f}_t(x) + \alpha \tilde{f}_t(y).
\]

When \(0 = \tilde{b}_t \leq x, y \leq \tilde{B}_t\) or \(x, y \geq \tilde{B}_t\), the inequality (21) is obtained by the concavity of linear functions and constant functions, respectively. The only remaining case is when \(\tilde{b}_t \leq x \leq \tilde{B}_t\) and \(y \geq \tilde{B}_t\), since \(x, y\) are interchangeable. Then, \(\tilde{f}_t(x) = B_t\) and \(\tilde{f}_t(y) = B_t \leq y\), thus if \((1-\alpha)x + \alpha y > B_t\),

\[
\tilde{f}_t((1-\alpha)x + \alpha y) = B_t = (1-\alpha)B_t + \alpha \tilde{f}_t(y) \geq (1-\alpha)\tilde{f}_t(x) + \alpha \tilde{f}_t(y);
\]

otherwise \((1-\alpha)x + \alpha y \leq B_t\),

\[
\tilde{f}_t((1-\alpha)x + \alpha y) = (1-\alpha)x + \alpha y = (1-\alpha)\tilde{f}_t(x) + \alpha \tilde{f}_t(y).
\]

Then, by the properties of composite concave functions, the objective function \(F(\cdot)\) is concave because both \(f_t(\cdot)\), \(\forall t\), and the linear function \(h(x) = W^T x\) are concave and nondecreasing.

Furthermore, if condition [5] holds true, it is straightforward to show that the objective is Lipschitz continuous by noting that \(0 \leq f_t(x) \leq x\), \(\forall t\) and potential value \(x\). Practically, the Lipschitz continuity can also be proven by reducing the objective function to be a finite sum up to \(t = T^\epsilon\), the number of time steps required for convergence of the underlying diffusion process with tolerance \(\epsilon > 0\). Meanwhile, the condition of the lower bounds, i.e. \(b_t \leq b_0 w^t\) where \(w = \min\{W_{ij} : W_{ij} > 0\}\), could be looser in practice, since, for example, the diffusion does not necessarily go through the edges of the smallest weight from the very beginning. Therefore, we can have a larger region of the parameters, where the objective function is (Lipschitz) continuous and concave.

When the objective function is concave with respect to the continuous variables, any local maximum is a global maximum, therefore the direct search algorithms can have global convergence. But we should also notice that this global convergence is only with respect to the continuous variables, since the optimality of the integer part is still local, as in definition 4.1. Meanwhile, when the objective function is Lipschitz continuous with respect to the continuous variables, there are methods proven to have global convergence, for example the new derivative-free line-search type algorithms [19].

The other case is when not only \(\{b_i\}\) are sufficiently small but \(\{B_i\}\) are sufficiently large, i.e. the linear-dynamics extreme as in section 3.2, where \(f_t(x) = x\), \(\forall t\). By the general diffusion model (2), the state values at each time step \(t > 0\) can be exactly computed as follows.

\[
x(t) = f_t(W^T x(t - 1)) = W^T x(t - 1) = (W^T)^t x(0).
\]
Therefore, the objective function in (17) is then,
\[ F(x(0)) = \sum_{t=1}^{\infty} 1^T x(t) = \sum_{t=1}^{\infty} 1^T (W^T)^t x(0) = 1^T \left( (I - W^T)^{-1} - I \right) x(0) = c^T x(0), \] (22)
where \( c = ((I - W)^{-1} - I)1 \), \( I \) is the identify matrix, and the penultimate equation is obtained given that condition [5] is true in the linear-dynamics extreme. Hence, the objective function is linear, thus (Lipschitz) continuous, concave and smooth, with the first-order derivative with respect to \( x(0) \) being \( c \).

The exact solution in this case is achievable in theorem 4.3.

**Theorem 4.3.** When \( \{b_t\}, \{B_t\} \) are as in theorem 3.2, then the exact solution to the MINLP in (19) is to choose \( x_i = B_0 \) and \( z_i = 1 \) if \( i \in A = \{i_1, \ldots, i_K\} \) s.t. \( \hat{c}_j \leq \hat{c}_l \), \( \forall j \notin A, l \in A \), where \( \tilde{c} = (I - W)^{-1}1 \) is the Katz centrality, and \( x_i = z_i = 0 \) otherwise.

**Proof.** By theorem 3.2, the general diffusion model reaches the linear-dynamics extreme. Then as in (22), the objective function is linear in its continuous variables, \( F(x) = c^T x, \) where \( c = ((I - W)^{-1} - I)1 \). Then if we fix the integer variables \( z \) and analyse the MINLP (19) with respect to the continuous variables \( x \), it can then be decomposed into \( N \) sub-problems,
\[ \max_{x_i} F_i(x_i) = c_i x_i \]
\[ \text{s.t.} \quad x_i \leq B_0 z_i, \]
\[ x_i \geq b_0 z_i, \]
\[ x_i \in \mathbb{R}. \] (23)

It is straightforward that the optimal solution to each sub-problem corresponding to node \( v_i \) is \( x_i = B_0 z_i \), since \( c_i, z_i \geq 0, \forall i \), with the optimal value \( F_i^*(x_i) = c_i B_0 z_i \). Then we consider the remaining problem with respect to the discrete variables,
\[ \max_{z} \sum_i F_i^*(x_i) = \sum_i c_i B_0 z_i \]
\[ \text{s.t.} \quad \sum_i z_i \leq K, \]
\[ z_i \in \{0, 1\}, \forall i. \] (24)

It is then apparent that the optimal solution is to set \( z_i = 1 \) if node \( i \) is ranked among the top \( K \) according to the value \( c_i B_0 \), or equivalently \( c_i \) since \( B_0 \) is a constant.

By definition,
\[ c = ((I - W)^{-1} - I)1 = (I - W)^{-1}1 - 1 = \tilde{c} - 1, \]
thus \( c_i = \tilde{c}_i - 1 \), and ranking each node \( v_i \) by \( c_i \) or \( \tilde{c}_i \) are equivalent.

Hence, as proved in theorem 4.3, the exact solution to the general diffusion model in the linear-dynamics extreme is to activate the \( K \) nodes of the highest Katz centrality, and set the initial value to be the upper bound \( B_0 \). This connects the IM problem with the general diffusion model to a well-studied centrality measure in networks, the Katz centrality. Further, this result can also be helpful in the general IM problem, since the solution can serve as a warm start for the following search algorithm, with the search depth potentially proportional to the distance of the underlying diffusion from this special case, i.e. the linear-dynamics extreme.
4.4 Customised direct search

Here, we exploit one feature of the objective function, that \( F(x) \) is non-decreasing in \( x \), which is inherited in the proof of theorem 4.3, and accordingly propose a customised direct search method for the IM problem with the general diffusion model.

To see why \( F(x(0)) \) is non-decreasing in \( x(0) \), we first show that \( x(t) \) is non-decreasing in \( x(0) \), \( \forall t > 0 \). This can be proven by induction: (i) \( x(1) \) is non-decreasing in \( x(0) \), since both \( f_1(\cdot) \) and the linear function \( h(x) = W^T x \) in (2) are non-decreasing; (ii) suppose \( x(t) \) is non-decreasing in \( x(0) \), \( \forall t \leq t' \), then we can show that \( x(t' + 1) \) is non-decreasing in \( x(0) \), since \( x(t' + 1) \) is non-decreasing in \( x(t') \) by the same logic as (i) and \( x(t') \) is non-decreasing in \( x(0) \) by the induction hypothesis. Hence, \( F(x(0)) = \sum_{t=1}^{\infty} 1^T x(t) \) is non-decreasing in \( x(0) \).

Accordingly, to maximise the objective \( F(x) \) with respect to \( x \) and \( z \) in the MINLP (19), is equivalent to the maximisation with \( x \) thus \( z \) at their highest possible values, particularly \( x = B_0 z \) and \( \sum_i z_i = K \), i.e.

\[
\max_z \quad F(B_0 z) \\
\text{s.t.} \quad \sum_i z_i = K, \quad z_i \in \{0, 1\}, \forall i. 
\]

(25)

Then the domain is a natural mesh to search at each iteration \( k \),

\[
M_k = \Omega^d = \{ z \in \{0, 1\}^n : \sum_i z_i = K \}. 
\]

(26)

We define the neighbourhood function of binary variables \( z \) to be,

\[
\mathcal{N}(z) = \{ y \in \{0, 1\}^n : \| y - z \|_1 \leq C \}, 
\]

(27)

where \( C \in \mathbb{Z}^+ \setminus \{1\} \), since \( \| y - z \|_1 \geq 2 \) if \( y \neq z \) and \( y, z \in \Omega^d \), where the shortest distance of 2 occurs when exchanging only one element of value 1 with another of value 0.

We then propose the following customised direct search (CDS) algorithm for the revised MINLP (25).

In this algorithm, we search the local neighbourhood of the current candidate \( z^{(k)} \) at each iteration \( k \) in the poll step, until a point with sufficient improvement in the objective value has been found or all points have been exhausted. In the termination check step, if an improved point has been found, the algorithm will go back to the poll step, but will decrease the required improvement if a sufficiently improved point has not been found, while if no improvement has been found, the algorithm terminates; see algorithm 3 for more details. The default parameter values are set to be \( \zeta = 0.1, \delta = 0.5 \) and \( C = 2 \).

Algorithm 3 Customised direct search (CDS).

1. Initialization: Set \( 0 < \zeta, \delta < 1 \). Let \( z^{(0)} \in \Omega^d \) such that \( z_i^{(0)} = 1 \) if node \( i \in A = \{i_1, \ldots, i_K\} \), s.t. \( \tilde{c}_j \leq \tilde{c}_l, \forall j \notin A, l \in A \), where \( \tilde{c} = (I - W)^{-1} 1 \) is the Katz centrality. Set iteration \( k = 0 \).

2. SEARCH step (optional): Evaluate \( f_{\Omega^d} \) on a finite subset of trial points on the mesh \( M_k \) (26), until a sufficiently improved mesh point \( z \) is found, where \( f_{\Omega^d}(z) > (1 + \zeta)f_{\Omega^d}(z^{(k)}) \), or all points have been exhausted. If an improved point is found, then the SEARCH step may terminate, skip the next POLL step and go directly to step 3.

3. POLL step: Evaluate \( f_{\Omega^d} \) on the set \( \Omega^d \cap \mathcal{N}(z^{(k)}) \subset M_k \) as in (27), until a sufficiently improved mesh point \( z \) is found, where \( f_{\Omega^d}(z) > (1 + \zeta)f_{\Omega^d}(z^{(k)}) \), or all points have been exhausted.

4. Termination check: If an improvement is found, set \( z^{(k+1)} \) as the improved solution, while decrease \( \zeta \leftarrow \delta \zeta \) if a sufficient improvement has not been found, increment \( k \leftarrow k + 1 \), and go to step 1. Otherwise, output the solution \( z^{(k)} \).

Therefore, local convergence is directly guaranteed in the termination step, by definition (4.1). Global convergence could possibly be obtain with a sophisticatedly developed SEARCH step and a better understanding of the landscape of the objective function, in order not to be trapped in bad local optima. However, the
downside of a global method is its time complexity, thus we leave the search step optional. Instead, the CDS method incorporates the features of the problem by using the exact solution to the linear-dynamics extreme, as in theorem 4.3 in the initialisation, and we postulate that the local optima near this special solution is sufficiently good.

From the current CDS method, there are at least two dimensions for further improvement in the quality of the output. We first note that the current problem is only of binary variables, thus is purely combinatorial, i.e. the problem is which set of nodes we should choose to give value 1. Accordingly, there are two known methods of global convergence: (i) brute-force, where each set of nodes of size $K$ are evaluated in order to choose the one of the largest value; (ii) random sampling, where sets of nodes are chosen randomly in all possible sets, and this method has global convergence asymptotically if it samples densely enough. The two dimensions of improvement are motivated by these two methods. On the one hand, we can enlarge the distance in defining the neighbourhood, which necessarily search more points in the domain. Further, if the neighbourhood is as large as the whole domain, it reduces to the brute-force method. On the other, we can restart the searching process, i.e. steps 1,2,3 in algorithm 3 from other unexplored points randomly, which works in the same logic as the search step. The strategy will give global convergence asymptotically, similar to the random sampling method.

The time complexity of the CDS method is composed of two parts (I) the evaluation of the objective function, and (II) the number of evaluations required until convergence. From Theorem 4.1, we know the complexity of evaluating the objective function, where given a network, it has the largest value when the diffusion model is in the linear-dynamics extreme. Therefore, the CDS method circumvents the worst-case complexity by starting from the solution associated to the linear-dynamics extreme. Further, (II) can be further decomposed into the product of (i) the size of discrete neighbourhood that is feasible, and roughly, (iia) the number of steps towards convergence, or more precisely, (iib) the number of evaluations divided by the neighbourhood size. (i) is $K(N - K)$, by (26) and (27), and is equivalently $O(N)$ since $K$ is normally assumed to be constant, i.e. $O(1)$. The only remaining part for a full description of the time complexity is (iia) or (iib), both are related to the rate of convergence which is difficult to provide a theoretical guarantee given the general properties of our objective function. Instead, we conjecture that the complexity of (iib) does not increase significantly when varying the network size $N$, i.e. at most $O(1)$, and verify it empirically through experiments (see appendix C for details). These all together result in the value of (II) to be $O(N)$. To measure the goodness of this complexity value, we compare it with a global algorithm, the brute-force method, where each node set will be evaluated. Then, overall $\binom{N}{K}$ evaluations are needed, which is proportional to $O(N^K)$ since again $K = O(1)$. Therefore, $O(N^{K-1})$ more multiples of evaluations are needed in the global algorithm than the CDS method. As in the following section 5, the complexity of the CDS is approximately linear in $N$, while that of brute-force has approximately exponential increase with respect to $N$.

5 Numerical experiments

In this section, we experimentally illustrate the features of the general class of information diffusion model, particularly through the differences from the IC model and the LT model after appropriate extensions as in section 3.2. We then turn to the features of the IM problem, and evaluate the performance of the CDS method in both small and large, both synthetic and real networks. Throughout the section, $b_0 = B_0 = 1$, and we apply exclusively the threshold-type bounds in (12), thus the upper and lower bounds vary according to the upper and lower bound thresholds, $a_B$ and $a_b$, respectively.

5.1 Performance evaluation of the diffusion model

We start from the general class of information diffusion model. In the following, we will numerically illustrate its rich behaviour when changing the upper and lower bounds, thus gradually deviating from the two extremes corresponding to the IC model and the LT model, as in sections 3.2 and 3.3 respectively.

Specifically, we consider simple networks with community structure. The networks are generated from the two-block planted $SBM(0.9, 0.1)$ where an edge is placed between the nodes in the same community with probability $p_{in} = 0.9$ and in the different communities with $p_{out} = 0.1$. We choose these values to
Figure 2: One realisation of the two-block planted $SBM(0.9, 0.1)$.

construct networks of assortative communities, and also to have a large difference between node sets in the same community and those not, for visualising purpose. The networks have size $N = 50$ and $N_c = 2$ communities, where we label the nodes in communities one and two as 0 to 24 and 25 to 49, respectively; see figure 2 for one realisation. We assign a uniform weight $\alpha = 0.1$, to account for general loss of information during transmission. Therefore, the lower bound threshold $a_b = 1$ corresponds to the critical lower bounds for the linear-dynamics extreme, where any $a_b > 1$ cannot always result in a linear dynamics. The difference in the diffusion behaviour will be quantified by the diffusion profile, defined as the aggregated state value up to time step $t'$,

$$\tilde{F}(t'; \mathbf{x}(0)) = \sum_{t=0}^{t'} \sum_{i=1}^{N} x_i(t),$$

changing along $t'$, where $\mathbf{x}(0) = (x_i(0))$ is the given initial state vector. Note that $\tilde{F}(t'; \mathbf{x}(0)) - \sum_{i=1}^{N} x_i(0)$ is part of the objective (17) in the IM problem, and the initial aggregated states are added for visualising the state changes from the beginning.

We first test the threshold effect imposed by increasing the lower bounds from the linear-dynamics extreme. Specifically, we compare the diffusion profiles from the following two different initially activated node sets: (i) $\{0, 1\}$ from the same community; (ii) $\{0, 25\}$ from the different communities. As discussed in section 3.2.1, whether the initially activated nodes are in the same community is not expected to make a difference when the underlying dynamics is perfectly linear, but nodes in the same community will immediately trigger a substantially higher diffusion once the lower bound threshold exceeds the critical value. This is also numerically clear from the comparison between the linear-dynamics extreme, $a_b = 1$, and the general diffusion model with a slightly higher value, $a_b = 2$, while the upper bound threshold $a_B$ being sufficiently large; see figure 3 for the results from 1000 samples of the SBM. The performance of set (ii) chosen from the two different communities varies more from sample to sample, because it largely depends on the inter-community edges whose existence has much smaller probability. Further investigation indicates that the diffusion profiles from set (ii) are concentrated at the values slightly above the mean (not as high as the mean from set (i)), while it also contains values substantially lower than the mean, together causing the high variance.

We then examine the locally linear effect added as the upper bounds become larger from the linear-threshold extreme. Here, we choose $a_b = a_B = 2$, the smallest integer for the threshold effect to take place, for the linear-threshold extreme. Accordingly, we increase the number of initially activated nodes to be 4, and consider the diffusion from the following two sets: (i) $\{0, 1, 2, 3\}$ from the same community; (ii) $\{0, 1, 25, 26\}$ evenly distributed in the two different communities. As mentioned in section 3.3 the aggregated state value associated to set (i) has larger increase than set (ii) as the upper bounds rise. This

\footnote{Note that as long as the probabilities satisfy the condition that $p_{in} > p_{out}$, the similar phenomena will occur.}
is verified numerically from improving the upper bound threshold from $a_B = 2$ to $a_B = 4$ while maintaining the lower bound threshold $a_b = 2$; see figure 4 for the results from 1000 samples of the SBM. The higher increase from set (i) is because the increased upper bounds are more likely to be reached by activating nodes in set (i) than set (ii), and a node of substantially higher state value can transmit the diffusion to other nodes even though it is the only source, i.e. locally linear effect. Furthermore, we observe that the diffusion triggered by set (ii) is consistently higher than the other, and they can also have competing performance when the upper bounds are sufficiently large.

Finally, we combine the two aspects and analyse the effects while changing both the upper and lower bounds. Accordingly, we compare the diffusion initiated by the node sets $\{0, 1\}$ versus $\{0, 25\}$ and also $\{0, 1, 2, 3\}$ versus $\{0, 1, 25, 26\}$, in terms of the ratio of the aggregated state values in (17),

$$R(\mathcal{A}_0^{(1)}, \mathcal{A}_0^{(2)}) = F(x^{(1)}(0))/F(x^{(2)}(0)),$$

where $x^{(1)}(0), x^{(2)}(0)$ are the initial state vectors from the initially activated node sets $\mathcal{A}_0^{(1)}$ and $\mathcal{A}_0^{(2)}$ respectively. We observe that the previously analysed features generally exist for different values of upper and lower bound thresholds: (i) the node sets in the same community have consistently higher aggregated state value as the lower bound threshold $a_b$ exceeds the critical value for the other set, which is the number of nodes distributed in each community here, i.e. 1 for $\{0, 25\}$ and 2 for $\{0, 1, 25, 26\}$; (ii) the node sets in each pair have increasingly similar performance as the upper bound threshold $a_B$ rises, and the ratio is very close to 1 when $a_B$ is large; see figure 5 for the results averaged from 1000 samples of the SBM.
5.2 Performance evaluation of the influence maximisation

We now proceed to the IM problem. In the following, we will first explore the saturation properties of the optimal objective value with respect to the budget size \( K \), when changing the diffusion model parameters, \( a_b \) and \( a_B \). We then test the accuracy of the proposed CDS method on both real and synthetic networks in a relatively small scale. Finally, we examine its performance on relatively large networks, by comparison with other state-of-the-art approaches: (i) random sampling, (ii) degree-centrality method, and (iii) Katz-centrality method, as well as other related heuristics.

5.2.1 Effect of saturation

We illustrate the saturation characteristics of the optimal objective value with respect to the budget size, \( K \), on a simple network. Consider a network of \( N = 20 \) nodes, generated from a two-block SBM with connecting probability \( p_1 = 0.5 \) in one community, \( p_2 = 0.25 \) in the other, and \( p_{12} = 0.05 \) between the two communities. The probabilities in the two communities are set to be different in order to separate the nodes in each community. We again assign uniform weight \( \alpha = 0.1 \). Since varying the upper and/or lower bounds will change the absolute value of the objective, here we compare the relative optimal objective value, \( F^*/F_{\text{max}}^* \), where \( F_{\text{max}}^* \) is the optimal value from \( K = N = 20 \), i.e. the maximum objective value with respect to all possible \( K \).

We observe that the optimal objective value reaches its maximum level earlier, or at a smaller budget size \( K \), as the upper bounds decrease; see figure 6 for two different cases of the lower bound threshold, \( a_b = 1 \) (left) and \( a_b = 2 \) (right).

![Figure 5](image5.png)

Figure 5: The ratios of the aggregated state values in \( \{0, 1\}/\{0, 25\} \) (left) and \( \{0, 1, 2, 3\}/\{0, 1, 25, 26\} \) (right), with changing upper bound (x-axis) and lower bound (y-axis) thresholds, on 1000 samples of \( \text{SBM}(0.9, 0.1) \).

![Figure 6](image6.png)

Figure 6: The change of relative optimal objective value with respect to the budget size \( K \) (x-axis) from the SBM, when the general diffusion model has increasing upper bound threshold \( a_B \) in the two cases of lower bound threshold: \( a_b = 1 \) (left) and \( a_b = 2 \) (right).
Figure 7: The change of relative optimal objective value with respect to the budget size $K$ (x-axis) from the SBM, when the general diffusion model has increasing lower bound threshold $a_b$ in the two cases of upper bound threshold: $a_B = 6$ (left) and $a_B = 7$ (right).

$a_b = 1$ and $a_b = 2$. When $a_b = a_B = 2$, only activating $K = 10$ nodes initially can achieve the maximum level of influence on the network, and for $a_b = a_B = 1$, only $K = 6$ nodes are needed. This property illustrates the rationality for the IM problem, where one aims to influence a large portion of the network from a small set of initially activated nodes: it is not only because of limited source, but also that activating more nodes does not necessarily benefit the optimisation substantially.

Interestingly, when enlarging the lower bounds, the early-saturation characteristics is not significant, but the step effect is increasingly explicit; see figure 7 for two separate cases of the upper bound threshold, $a_B = 6$ and $a_B = 7$. When $a_b = a_B$, either 6 or 7, the optimal objective function changes exactly as a step function with respect to the budget size $K$, and when $a_b = 1$, it is strictly increasing, close to being linear. When $a_b$ lies in between these two extremes, the optimal objective function interpolates these two shapes, with linear-like changes as well as certain step effect. The results further demonstrate that the general diffusion model interpolates between the linear dynamics and the LT model.

5.2.2 Performance of the CDS method

We now examine the CDS method. From section 4.4, we know that the CDS method is a local algorithm, thus can only guarantee convergence to a local maximum. Therefore, it is important to explore how good the output is with respect to the global optimum. However, algorithms with global convergence, such as the brute-force method, require $O(N^K)$ evaluations of the objective as illustrated in section 4.4 which prohibits their application to large networks. Hence, we exclusively consider networks in a relatively small scale, both real, e.g. the Karate club network, and synthetic, e.g. the SBM as in section 5.1.

In order to measure the goodness of the output, we consider the following two measures: accuracy and rank. The accuracy is defined as the relative value to the global optimum,

$$a(F^{res}; F^*) = F^{res}/F^*,$$

where $F^{res}$ is the algorithm output, and $F^*$ is the global optimum. $a(\cdot) \in [0, 1]$ and a higher accuracy implies a better solution. Since the current problem can be reduced to a combinatorial optimisation in terms of the initially activated node set, it is feasible to analyse the rank of the output set in all possible candidate sets, which complements the previous information. The rank is defined as

$$r(A_0^{res}; V, K) = \left( |\{A_0 \subset V : |A_0| = K, F(B_0 z_{A_0}) > F(B_0 z_{A_0^{res}})\}| + 1 \right)/|\{A_0 \subset V : |A_0| = K\}|,$$

where $G(V, E)$ is the underlying network, $K$ is the budget size, $A_0^{res}$ is the set of initially activated nodes in the output, and $z_{A_0}$ is the vector w.r.t the initially activated node set $A_0$. $r(\cdot) \in (0, 1]$, and a lower rank implies a better solution. Accordingly, the reference global algorithm we choose is the brute-force method, where the objective values corresponding to all node sets of size $K$ are computed.

In the following, we test the performance of the CDS method in the two occasions: (i) differentiating the upper and lower bound thresholds while maintaining the same budget size $K$, and (ii) varying $K$ while fixing the upper and lower bound thresholds. We again assign uniform weight $\alpha = 0.1$ in the networks.
Karate club network. The karate club network is a social network of a university karate club [44]. It captures $N = 34$ members of the club, and has $|E| = 78$ edges indicating pairs of members who interact outside the club. This real network is comprehensively used in network analysis for various purposes, while in a manageable small scale, thus is suitable for the testing here.

We first maintain the budget size $K = 3$, and test the performance of the CDS method when varying the lower bound threshold $a_b \in [0, 2]$ and the upper bound threshold $a_B \in [a_b, 9]$. The CDS method successfully finds an optimal solution with accuracy 1 and rank 0.017% in all different choices of lower and upper bound thresholds; see figure 8. As mentioned in the beginning, the CDS method is a local algorithm, and its performance depends on the distance between the initial point and the optimal solution(s). Therefore, the global convergence implies that searching from the nodes chosen by Katz centrality is one optimal strategy for the IM tasks on the karate club network7. Furthermore, one outstanding reason why local methods are interesting is its speed of convergence, and here the time consumed by the CDS method is always less than 5% of the brute-force’s which is between 1s and 4s.

Figure 8: Performance of the CDS method on the karate club network, with changing upper (x-axis) and lower (y-axis) bound thresholds of the general diffusion model and $K = 3$, in terms of the accuracy $a$ (left) and the rank $r$ (% right).

We then change the budget size $K$ in the following two combinations of lower and upper bound thresholds: (i) $a_b = 2 < a_B = 8$, and (ii) $a_b = 2 = a_B$, the linear-threshold extreme. The performance of the initial point of the CDS method, i.e. selecting nodes by Katz centrality, is given for reference. For case (i) when $a_b < a_B$, the CDS method can always find a global optimal solution while the performance of the initial points vary; see figure 9. For case (ii) when $a_b = a_B$, the CDS method can find a global optimal solution when $K \leq 5$ and the output still has accuracy $a > 0.97$ when $K$ is as large as 7, while the initial points have more variance in performance than the previous case when $K$ is large; see figure 10. Note that the number of node sets of size $K$, i.e. the denominator of the rank $r$, becomes $O(N)$ multiples as $K$ increases by a unit, thus the rank in figure 9 decreases approximately linearly in the log-linear plot, with log scale in the y-axis. The rank plot in figure 10 is close to the previous linear decrease, except one obvious deviation when $K = 6$ where the rank is roughly at the same level as $K = 5$, thus instead of finding the top 1 set, it outputs a top $(N - 6 + 1)/6 \approx 5$ set over all 1,344,904 possible sets. For the speed of convergence, the time consumption of the CDS method increases approximately linearly as $K$ rises, while for the brute-force, it changes exponentially, which makes it practically hard to obtain the global optimum under larger $K$.

The drop in accuracy when $K$ becomes larger in the case $a_b = 2 = a_B$ is partially because the initial points have increasingly worse performance, and also partially because the fixed neighbourhood size, i.e. 2 here, becomes relatively smaller. The former is apparent from figure 10 and is expected since the initial point is the exact optimal solution in the linear-dynamics extreme while the current case corresponds to the other end. The latter is inherited in the CDS method being local, since one needs to define the radius

7To explore the initial-state dependence of the direct search method, we apply the CDS method but starting from all other possible node sets, to the same tasks. For each pair of the bound thresholds, we observe that the method can still converge to a global optimum searching from approximately 40% of all such sets, which implies that the objective in the karate club network has a relatively large part of the domain that is concave-like.
of local neighbourhood. As discussed in section 4.3, there are two dimensions to further improve the algorithm: (i) to enlarge the radius of neighbourhood as \( K \) increases, e.g. an adaptive neighbourhood; (ii) to restart the search from some other points, preferably far from the original start.

**Stochastic block model (SBM).** The SBM under consideration is of the same size and weights as in section 5.1 but with different probabilities in the two blocks, \( p_1 = 0.3 \) and \( p_2 = 0.12 \), in order to distinguish the nodes in the two different communities, and the connecting probability between the two communities being \( p_{12} = 0.01 \). As analysed in section 5.1 when \( p_1 = p_2 > p_{12} \), node sets with different distributions in the two communities have separating diffusion profiles while changing the bound values. It is then interesting to explore the performance of the CDS method for the corresponding IM tasks on SBMs.

When the budget size is fixed at \( K = 4 \), the CDS method can find an optimal or fairly close to optimal solution when \( a_b \leq 1 \) and when \( a_B \) is far from \( a_b \); see figure 11. This is expected since both cases share certain features with the linear-dynamics extreme associated to the initial point, either \( a_b \) is sufficiently low or \( a_B \) is sufficiently high. As \( a_B \) gets closer to \( a_b \), the CDS method has increasingly worse behaviour, and the worst performance occurs in the linear-threshold extreme with \( a_b > 2 \). However, in such worst-case scenarios, the solutions still have accuracy around 0.77 and rank slightly above 0.01\% in overall 230,300 possibly initially activated sets, i.e. the CDS method can still output a top 30 set. This actually demonstrate the close-to-optimal performance of the CDS method for the IM tasks on here.

Since there is a small distance between the CDS output and a global optimum, we explore one improvement strategy here: to restart the search process, i.e. steps 1,2,3 in algorithm 3, from other unexplored points. To determine how to select such initial points, we approach it from the structural characteristics. Noticeably, there are two communities in the network. Accordingly, we propose the following community restart strategy: (i) construct a set containing different splits of \( K \) into the two communities, \( S := \{(K_1, K_2) : K_1 + K_2 = K, K_1, K_2 \in \mathbb{N}\} \); (ii) construct the set of initial points corresponding to activating \( K_1 \) and \( K_2 \) nodes of the highest Katz centrality in communities 1 and 2, respectively, for each split.
Figure 11: Performance of the CDS method on the SBM, with changing upper (x-axis) and lower (y-axis) bound thresholds of the general diffusion model and $K = 4$, in terms of the accuracy $a$ (left) and the rank $r$ (%, right).

Figure 12: Performance of the CDS method plus the community restart strategy on the SBM, with changing upper (x-axis) and lower (y-axis) bound thresholds of the general diffusion model and $K = 4$, in terms of the accuracy $a$ (left) and the rank $r$ (%, right).

$(K_1, K_2) \in S$; (iii) restart the search process from each set in (ii) if it is has not been explored yet. The community restart strategy assists the CDS method to find a global optimum when $K = 4$, with almost all different combinations of the bound thresholds; see figure [12]. Further investigation indicates that more than three quarters of all improvements are obtained from splitting $K$ according to, e.g. the ratio of the two communities’ densities, which implies potential redundancy in the splits considered.

We further explore the improvement from the community restart strategy on the CDS method when increasing the budget size $K$ and keeping the bound thresholds $a_b = 2$ and $a_B = 3$. The bounds are chosen to be one of the pairs where the CDS method with the community restart strategy can find a global optimal solution while the plain one cannot in the case of $K = 4$, for a competing performance to the best-case scenario’s. We observe that the plain CDS method cannot find a global optimal solution when $K < 5$, while the community restart strategy successfully push the method to reach a global optimum for all those values; see figure [13]. Further investigation shows that the improvements are still from the split close to the aforementioned ratio of the two communities’ densities. Hence, together with the further experiments in appendix B.4, we also postulate that the underlying split rule could be effectively reduced to be either all in one community or according to the ratio, or the inverse ratio, of their densities. Finally, on the one hand, this is a good-case scenario, and the community restart strategy could not improve the performance in certain situations, which implies the complexity of the optimisation problem; on the other hand, the plain CDS method already provides a reasonably good solution for the IM tasks here. For example, when $K = 2$, even though the accuracy is around 0.85, the rank is still as low as 0.3% in overall 1225 candidates, thus the CDS method can still find a top 4 set.8

*This emphasises the rationality to complement the accuracy with the rank in measuring the quality of an output, since in...

25
5.2.3 Comparison between methods

In the following experiments, we examine the performance of the CDS method, together with the community restart strategy, on both synthetic and real networks in a relatively large scale, and compare them with the state-of-the-art methods. Hereafter, we denote the set of initially activated nodes, $A_0$, as the seed set. Since the objective function (17) by the general diffusion model (2) lacks the important sub-modular property, the vast majority of algorithms following the work of Kempe et al [23] are not applicable here. We instead compare the CDS method with more network-structure-centred algorithms as follows.

(i) **Random sampling** ("Random"). Randomly selecting $K$ nodes of the network, and return them as the seed set. In the experiments by Kempe et al [23], it has been used as a baseline method. Here, we repeat the process $n$ times, and output the one with the highest objective value.

(ii) **Degree-centrality method** ("Degree"). Centrality is an important measure in network analysis, which measures how significant a node is in the network. There are many centrality-based methods proposed for the IM problem in the literature [5]. The degree-centrality method is to select the $K$ nodes of the highest degree in the network as the seed set.

(iii) **Katz-centrality method** ("Katz"). Finding the $K$ nodes of the highest Katz centrality in the network as the seed set.

Since searching more of the domain cannot worsen the performance (though the time complexity suffers), in order to provide a fair comparison with the CDS method plus the community restart strategy ("CDS-com"), we consider the following reference method.

(iv) **The CDS method with random restarts** ("CDS-rand"). Restarting the search process of the CDS method with a randomly selecting seed set, of the same times as the one with the community restart strategy.

Since both methods (i) and (iv) have random components, we will repeat the whole methods $n_r$ times and analyse their averaged performance. The corresponding objective value, $F$, is directly applied as the comparison measure.

The networks under consideration are composed of a large two-block SBM, and a real collaboration network. SBMs have the advantage that the ground-truth communities are known, which allows to examine the performance of the community restart strategy in a larger scale. Collaboration networks are extensively used in the IM experiments, because researchers believe that such networks capture many of the key features of social networks more generally [24]. The specific one we select also has ground-truth communities as metadata. In the following experiments, we choose $n = 100$ and $n_r = 10$. Since the network is now at a combinatorial optimisation, it is entirely possible that a solution does not have close-to-one accuracy but its objective is higher than almost all other feasible points, thus it can still be interpreted as being close-to-optimal.
relatively large scale, we apply the restricted version of the community restart strategy, according to the postulation in the last paragraph in section 5.2.2, where the extra starts are from the seed sets either with all nodes in one community or split with regard to the ratio, or the inverse ratio, of the two communities’ densities.

**Stochastic block model (SBM).** Consider a relatively large network of size $N = 1000$, generated by a two-block SBM with different probabilities in the two communities, $p_1 = 0.015$ and $p_2 = 0.005$, and the connecting probability between the two communities being $p_{12} = 0.0001$. These values are chosen to maintain roughly the same mean degree as the SBM in section 5.2.2, and we assign the same uniform weight $\alpha = 0.1$.

The CDS method outperforms all the reference algorithms in all possible budget sizes between 8 and 24; see figure 14 for the results from both a relatively hard region when $a_b = a_B = 8$ and a general case when $a_b = 4 < a_B = 8$. The performance is further improved by the community restart strategy, even in its restricted format, but not random starts. This observation verifies the valuable information in the community structure. The restriction rule could potentially be further tuned by incorporating more features of the problem, but we leave this to future work. Furthermore, the performance of the degree centrality and the Katz centrality is similar in both cases, and we can prove that the Katz centrality is expected to be proportional to the node degree in the SBMs (see appendix A).

The collaboration network analysed here is constructed by a comprehensive list of research papers in computer science provided by DBLP computer science bibliography, where two authors are connected if they publish at least one paper together [43]. An additional feature of this data is that there are intrinsic communities defined by metadata, that is characterised by the publication venue, e.g., journal or conference, which is useful for testing the performance of the community restart strategy in real cases. Accordingly, we select two such venues (no. 6035 and 6335) whose sizes are over 500 but with only 2 authors overlapping. The resulting network is unweighted and connected, containing $N = 1016$ nodes and $|E| = 3469$ edges, thus the average degree of nodes is 6.83. Since various factors may influence the community structure, not only publication venue, here we apply the ratio cut technique in spectral clustering to obtain empirical communities (or bipartition) [41]. The resulting communities largely agree the ground-truth ones, with both normalised mutual information and adjusted mutual information score as high as 0.889. Here, we assign a uniform weight $\alpha = 0.024$ in response to a higher largest eigenvalue of its adjacency matrix, which can guarantee the convergence of the corresponding linear dynamics.

Since the network contains several nodes of relatively high degrees which largely dominate the performance of the general diffusion model in the linear-dynamics extreme, both degree-centrality and Katz-

\[ \text{Figure 14: Objective value obtained from different node selection algorithms applied to the SBM, subject to changing budget size (x-axis), when the general diffusion model has } a_b = 4, a_B = 8 \text{ (left) and } a_b = a_B = 4 \text{ (right), where the shades of the same colors show the standard deviation of the corresponding random methods.} \]
Figure 15: Objective value obtained from different node selection algorithms applied to the collaboration network, subject to changing budget size (x-axis), when the general diffusion model has $a_b = 5, a_B = 10$ (left) and $a_b = a_B = 5$ (right), where the shade in purple shows the standard deviation of the ‘CDS-com’ method, and the results from random sampling is ignored because they are always close to 0.

Centrality methods perform competitively to the CDS method when $a_B$ is far from $a_b$. However, the CDS method still outperforms others, and the distance is larger as $a_B$ becomes smaller; see figure 15. There are still improvements from the community restart strategy in this real occasion, though not as significant as in the SBM before, but not random starts. Note that the modularity of the empirical communities is 0.481, slightly higher than the ground-truth communities 0.471, which indicates that there is certain modular structure of the underlying network but not very strong.

6 Conclusions

To understand how information diffuses through social networks has many practical implications. Among the vast amount of work in this field, there are two classic diffusion models with widely accepted mechanisms: the IC model and the LT model. However, the binary state variables, and no feedback mechanism in the diffusion processes are arguably restrictive. Therefore, we extend both classic models to address these issues, and further propose a novel class of diffusion model interpolating these extended models, where we show that our general diffusion model can be reduced to the extended IC model at one end, and the extended LT model at the other. Moreover, the general diffusion model goes far beyond simply connecting the two classic models, since, as illustrated in appendix B, it can have both types of dynamics coexist in a single network, which is not possible with any one of the classic diffusion models. This characteristics is clear from the experiments on a special network by connecting a regular lattice with a ER random graph by a few edges, where we see that the general diffusion model can successfully influence the whole network while the extended LT model can only influence part of it. We leave the investigation on real social networks to future work.

The general features of our proposed diffusion model necessarily complicate the corresponding IM problem. For example, the objective function is no longer submodular, a key property to obtain the theoretical approximation guarantee for various greedy algorithms. However, properties like submodularity are arguably strict, and breaking the boundary of such properties is necessary for the IM problem to embrace a wider family of diffusion models in practice [22,29]. Therefore, we propose a general framework for the IM problem, based on MINLP and derivative-free methods, where only evaluations of the objective function is required. With increasingly special properties, we can have better theoretical guarantees. Furthermore, we also propose the CDS method particularly for the IM problem with the general diffusion model. We note that the CDS method can be also applied to the classic IM problem, since both are to determine the set of nodes to activate at the start. We show through experiments that the CDS method can find a close-to-optimal solution, and also outperforms state-of-the-art centrality-based methods in both synthetic and real networks. Since the CDS method we proposed is in its plain format, one can further improve the performance by fine tuning each step as possible extensions.
In summary, we unify the features from the IC model and the LT model into a novel class of information diffusion model, and propose a general framework for the IM problem that is applicable to a broad range of functions describing the overall influence. As the mechanisms behind the two classic models are widely accepted in information diffusion, we believe the proposed general diffusion model has the potential to explain more diffusion phenomena on, but not restricted to, social networks. Meanwhile, the IM framework we propose in the present paper provides a systematic approach to handle the IM problem when the objective function does not have desired properties (e.g., submodularity), which leaves more freedom to design various diffusion models and provides insights into solving the IM problem in more realistic occasions.

Acknowledgements

We thank Sebastian Lautz, Alisdair Wallis, Karel Devriendt, and Lindon Roberts for useful discussions.

References

[1] M. Abramson, C. Audet, J. Chrissis, and J. Walston. Mesh adaptive direct search algorithms for mixed variable optimization. *Optim. Lett.*, 3:35–47, 2009.

[2] M. Asllani, T. Carletti, F. Di Patti, D. Fanelli, and F. Piazza. Hopping in the crowd to unveil network topology. *Phys. Rev. Lett.*, 120:158301, 2018.

[3] C. Audet, Le Digabel S., V. Montplaisir, and C. Tribes. NOMAD version 4: Nonlinear optimization with the MADS algorithm. *preprint*, arXiv:2104.11627v2, 2021.

[4] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st International Conference on World Wide Web*, pages 519–528. ACM, 2012.

[5] S. Banerjee, M. Jena, and M. Pratihar. A survey on influence maximization in a social network. *Knowledge Inf. Syst.*, 62:3417–3455, 2020.

[6] P. Belotti, C. Kirches, S. Leyffer, J. Luedtke, and A. Mahajan. Mixed-integer nonlinear optimization. *Acta Numer.*, 22:1–131, 2013.

[7] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 946–957. SIAM, 2014.

[8] F. Boukouvala, R. Misener, and C. Floudas. Global optimization advances in Mixed-Integer Nonlinear Programming, MINLP, and Constrained Derivative-Free Optimization, CDFO. *European J. Oper. Res.*, 252:701–727, 2016.

[9] A. Bovet and H. Makse. Influence of fake news in twitter during the 2016 us presidential election. *Nat. Commun.*, 10(7), 2019.

[10] Samuel Burer and Adam N. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surv. Oper. Res. Manag. Sci.*, 17:97–106, 2012.

[11] D. Centola. The spread of behavior in an online social network experiment. *Science*, 329:1194–1197, 2010.

[12] D. Centola and M. Macy. Complex contagions and the weakness of long ties. *Amer. J. Sociol.*, 113(3):702–734, 2007.

[13] N. Chen. On the approximability of influence in social networks. *SIAM J. Discrete Math.*, 23:1400–1415, 2009.
[14] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1029–1038. ACM, 2010.

[15] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In 2010 IEEE International Conference on Data Mining, pages 88–97. IEEE, 2010.

[16] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch. Mixing beliefs among interacting agents. Adv. Complex Syst., 03(01n04):87–98, 2000.

[17] M. Degroot. Reaching a consensus. J. Amer. Statist. Assoc., 69(345):118–121, 1974.

[18] D. Fanelli and A. McKane. Diffusion in a crowded environment. Phys. Rev. E, 82:021113, 2010.

[19] T Giovannelli, G Liuzzi, S Lucidi, and F Rinaldi. Derivative-free methods for mixed-integer nonsmooth constrained optimization. preprint, arXiv:2107.00601, 2021.

[20] A. Goyal, W. Lu, and L. Lakshmanan. Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In Proceedings of the 20th International Conference Companion on World Wide Web, pages 47–48. ACM, 2011.

[21] D. Guilbeault, J. Becker, and D. Centola. Complex contagions: A decade in review. In Complex Spreading Phenomena in Social Systems: Influence and Contagion in Real-World Social Networks, pages 3–25. Springer International Publishing, Cham, 2018.

[22] F. Gursoy and D. Gunnecc. Influence maximization in social networks under deterministic linear threshold model. Knowledge-Based Systems, 161:111–123, 2018.

[23] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 137–146. ACM, 2003.

[24] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. Theory Comput., 11:105–147, 2015.

[25] M. Laguna, F. Gortázar, M. Gallego, A. Duarte, and R. Martí. A black-box scatter search for optimization problems with integer variables. J. Global Optim., 58:497–516, 2014.

[26] S. Le Digabel. Algorithm 909: Nomad: Nonlinear optimization with the mads algorithm. ACM Trans. Math. Software, 37, 2011.

[27] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. ACM Trans. Web, 1:5–43, 2007.

[28] J. Leskovec, A. Krause, C. Guestrin, Christos Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 420–429. ACM, 2007.

[29] Y. Li, J. Fan, Y. Wang, and K. Tan. Influence maximization on social graphs: A survey. IEEE Trans. Knowledge Data Engyr., 30:1852–1872, 2018.

[30] Z. Lu, W. Zhang, W. Wu, J. Kim, and B. Fu. The complexity of influence maximization problem in the deterministic linear threshold model. J. Comb. Optim., 24:374–378, 2012.

[31] Elchanan M. and Roch S. Submodularity of influence in social networks: From local to global. SIAM J. Comput., 39:2176–2188, 2010.

[32] H. Ma, H. Yang, M. Lyu, and I. King. Mining social networks using heat diffusion processes for marketing candidates selection. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, pages 233–242. ACM, 2008.
A Stochastic block model

In this section, we discuss the aforementioned properties of stochastic block models (SBMs) in more detail, including the increase in the aggregated state values when changing the upper bounds and the relationship between degree centrality and Katz centrality. Here, we exclusively focus on two-block SBMs.

Take the same planted two-block SBM as in section 3.2.2 for example. We first show that when increasing the upper bounds, the increase in the aggregated state value, after one step, is expected to be larger when the initially activated node sets are in the same community rather than evenly distributed in the two communities, as mentioned in section 3.3.2.

Claim A.1. With \( b_0 = B_0 \), and \( b_1 = 2\alpha b_0 \), if \( \text{SBM}(p_{\text{in}}, p_{\text{out}}) \) of two equally-sized communities, \( B_1, B_2 \) of size \( N_b \), and uniform weight \( \alpha \) has

\[
p_{\text{in}} > p_{\text{out}}, \quad (1 - p_{\text{in}}) > p_{\text{out}},
\]

then when \( B_1 \) rises from \( b_1 = b_0 = 2\alpha b_0 \) to \( B_1 = 2b_1 = 4\alpha b_0 \) (with \( 4\alpha < 1 \)), the increase in the expected aggregated state value, \( \mathbb{E} \left[ \sum_i x_i(1) \right] \), from the initially activated node set (i) \( A_0 = \{v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}\} \subset B_1 \), is larger than that from (ii) \( A_0 = \{v_{j_1}, v_{j_2}, v_{j_3}, v_{j_4} : v_{j_1}, v_{j_2} \in B_1, v_{j_3}, v_{j_4} \in B_2\} \).

Note the same results can be obtained with arbitrary community sizes, but extra conditions of both the community sizes and the probabilities are required.
Proof. For the SBM, \( \mathbf{W} = \alpha \mathbf{A} \), where \( \mathbf{A} \) is the (unweighted) adjacency matrix. We maintain the same notations for the block membership of each node \( v_i, \sigma_i \in \{1, 2\} \), and the linear part of the state vector, \( y(t) = \mathbf{W}^T \mathbf{x} \), as in section 3.2.2.

In case (i),

\[
y_i(1) = \alpha b_0 (Bin(4, p_{in}) \delta(\sigma_i, 1) + Bin(4, p_{out}) \delta(\sigma_i, 2)),
\]

while in case (ii),

\[
y_i(1) = \alpha b_0 (Bin(2, p_{in}) + Bin(2, p_{out})).
\]

When \( B_1 = b_1 = 2\alpha b_0 \),

\[
\mathbb{E} \left[ \sum_i x_i(1) \right] = \sum_i 0P(y_i(1) < b_1) + 2\alpha b_0 P(y_i(1) \geq 2\alpha b_0).
\]

While when \( B_1 = 4\alpha b_0 \),

\[
\mathbb{E} \left[ \sum_i x_i(1) \right] = \sum_i (0P(y_i(1) < b_1) + 2\alpha b_0 P(y_i(1) = 2\alpha b_0) + 3\alpha b_0 P(y_i(1) = 3\alpha b_0) + 4\alpha b_0 P(y_i(1) = 4\alpha b_0))
\]

Hence, the increase in the expected aggregated state value is \( \Delta = (31) - (30) \), i.e.

\[
\Delta = \sum_i \alpha b_0 P(y_i(1) = 3\alpha b_0) + 2\alpha b_0 P(y_i(1) = 4\alpha b_0).
\]

Hence, in case (i),

\[
\Delta_{(i)} = \sum_i \alpha b_0 \left( \binom{4}{3} p_{in}^3 (1 - p_{in}) \delta(\sigma_i, 1) + \binom{4}{3} p_{out}^3 (1 - p_{out}) \delta(\sigma_i, 2) \right) + 2\alpha b_0 \left( p_{in}^4 \delta(\sigma_i, 1) + p_{out}^4 \delta(\sigma_i, 2) \right)
\]

\[
= N_b \times \alpha b_0 \times 2 \left( 2p_{in}^3 (1 - p_{in}) + 2p_{out}^3 (1 - p_{out}) + p_{in}^4 + p_{out}^4 \right).
\]

While in case (ii),

\[
\Delta_{(ii)} = \sum_i \alpha b_0 \left( \binom{2}{1} p_{in} (1 - p_{in}) p_{out}^2 + \binom{2}{1} p_{out} (1 - p_{out}) \right) + 2\alpha b_0 \left( p_{in}^2 p_{out}^2 \right)
\]

\[
= 2N_b \times \alpha b_0 \times \left( 2p_{in}^2 p_{out} (1 - p_{out}) + 2p_{out}^2 p_{in} (1 - p_{in}) + 2p_{in}^2 \right).
\]

Hence,

\[
\Delta_{(i)} - \Delta_{(ii)} = 2N_b \alpha b_0 \left( 2 \left( p_{in}^2 - p_{out}^2 \right) p_{in} (1 - p_{in}) - p_{out} (1 - p_{out}) + \left( p_{in}^2 - p_{out}^2 \right)^2 \right),
\]

which is positive given condition (29). \( \square \)

We now illustrate that the expected Katz centrality is proportional to the expected degree centrality in a general two-block SBM, where the probabilities in the two communities and their sizes can be different.

**Theorem A.1.** In a two-block SBM with the connecting probabilities in the two communities, \( B_1, B_2 \), being \( p_1, p_2 \), respectively, and between the two being \( p_{12} \), if a node \( v_i \) is expected to have higher degree centrality than another node \( v_j \),

\[
\mathbb{E} \left[ \sum_l A_{il} \right] > \mathbb{E} \left[ \sum_l A_{jl} \right],
\]

(32)
where \( A = (A_{ij}) \) is the adjacency matrix, then node \( v_i \) is expected to have higher Katz centrality than node \( v_j \),

\[
E \left[ \sum_{k=1}^{\infty} \alpha_{katz}^k \sum_{l} A_{li}^k \right] > E \left[ \sum_{k=1}^{\infty} \alpha_{katz}^k \sum_{l} A_{lj}^k \right],
\]

where \( \alpha_{katz} \) is the discounting factor.

**Proof.** In such SBM, for each pair of nodes \( v_i, v_j \), \( A_{ij} \) is an independently distributed Bernoulli random variable, with success probability \( p_1 \) if \( v_i, v_j \in B_1 \), \( p_2 \) if \( v_i, v_j \in B_2 \), and \( p_{12} \) otherwise. Hence, nodes in the same communities are equivalent, and there are only two distinct expected values in both centralities, one for each community.

For the degree centrality\[11\]

\[
E [\sum_{l} A_{li}] = \begin{cases} N_1 p_1 + N_2 p_{12}, & v_i \in B_1, \\ N_1 p_{12} + N_2 p_2, & v_i \in B_2, \end{cases}
\]

where \( N_1, N_2 \) are the sizes of communities \( B_1, B_2 \), respectively. Hence, condition (32) can only happen when nodes \( v_i \) and \( v_j \) are in different communities.

Without loss of generality, we assume \( v_i \in B_1 \), and then \( v_j \in B_2 \). We show (33) holds true by the following stronger relationship that for each \( k > 0 \),

\[
E \left[ \sum_{l} A_{li}^k \right] > E \left[ \sum_{l} A_{lj}^k \right].
\]

We prove it by induction on \( k \). (i) When \( k = 1 \),

\[
E \left[ \sum_{l} A_{li} \right] = N_1 p_1 + N_2 p_{12} = E \left[ \sum_{l} A_{lj} \right],
\]

where the inequality is by condition (32). (ii) Suppose (34) is true for all \( k \leq k' \). Then when \( k = k' + 1 \),

\[
E \left[ \sum_{l} A_{li}^{k'+1} \right] = E \left[ \sum_{l} \sum_{h} A_{li}^{k'} A_{hi} \right] = \sum_{v_h \in B_1} \sum_{l} \sum_{h} A_{li}^{k'} p_1 + \sum_{v_h \in B_2} \sum_{l} \sum_{h} A_{li}^{k'} p_{12} = E \left[ \sum_{l} A_{li}^{k'} \right] N_1 p_1 + E \left[ \sum_{l} A_{lj}^{k'} \right] N_2 p_{12},
\]

(35)

where the second equality is by independence, and the last equality is by equivalence among nodes in the same communities. Similarly,

\[
E \left[ \sum_{l} A_{lj}^{k'+1} \right] = E \left[ \sum_{l} A_{lj}^{k'} \right] N_1 p_{12} + E \left[ \sum_{l} A_{lj}^{k'} \right] N_2 p_2.
\]

Hence, (35) - (36) is

\[
E \left[ \sum_{l} A_{li}^{k'} \right] N_1 (p_1 - p_{12}) + E \left[ \sum_{l} A_{lj}^{k'} \right] N_2 (p_{12} - p_2) > E \left[ \sum_{l} A_{lj}^{k'} \right] (N_1 (p_1 - p_{12}) + N_2 (p_{12} - p_2)) > 0,
\]

where the first inequality is by induction hypothesis, and the last inequality is by condition (32).

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]
Further features of the diffusion model

In this section, we further discuss the features of the general diffusion model (2). We show that both the locally linear-dynamics-like and the locally linear-threshold-like diffusion can co-exist in a single network in section B.1. Moreover, we analyse the change of aggregated state value with respect to the initial state vector through the derivative information in section B.2, which plays an important role for the influence maximisation (IM) problem that we consider in section 4. Finally, we numerically illustrate the feature of coexistence of regimes in section B.3, and test the performance of the customised direct search (CDS) method for the corresponding IM tasks in section B.4.

B.1 Coexistence of regimes

As discussed in sections 3.2 and 3.3, the general diffusion model connects the IC model and the LT model after appropriate extensions. Note that the two dynamics have distinct characteristics, as the linear dynamics can diffuse through tree graphs efficiently, while the LT model can only influence the nodes with a sufficient number of influenced neighbours, typically greater than 1. Interestingly, we will show that both types of diffusion can co-exist in a single network with the general diffusion model.

![Figure 16: An example network with both tree-like substructure and regular-lattice-like substructure.](image)

Table 1: The state values of selected nodes in the first 3 steps from the general diffusion model with $a_b = 2$ and $a_B = 8$ on the network in figure 16 where the initially activated nodes are in red.

| Node | $t = 1$ | $t = 2$ | $t = 3$ |
|------|---------|---------|---------|
| 1    | $8a_b b_0$ | 0       | $8a^2 b_0$ |
| 2    | 0       | $8a^2 b_0$ | 0       |
| 3    | 0       | 0       | $8a^3 b_0$ |
| 12   | 0       | 0       | 0       |
| 13   | $2a_b b_0$ | 0       | $32a^3 b_0$ |
| 14   | $2a_b b_0$ | 0       | $32a^3 b_0$ |
| 15   | $2a_b b_0$ | 0       | $32a^3 b_0$ |
| 17   | 0       | 0       | 0       |
| 18   | 0       | $4a^2 b_0$ | 0       |
| 19   | 0       | $4a^2 b_0$ | 0       |
| 20   | 0       | $4a^2 b_0$ | 0       |

Note that the general diffusion going through the tree-like structure can only occur surrounding a node of large state value within limited number of time steps, until the threshold effect starts to function. For example, in figure 16 node $v_1$ hits the upper bound at $t = 1$, and the diffusion can further proceeds to $v_2$ and $v_3$ in the subsequent two steps, but not any further. Specifically, for a node has the upper bound as its state value at time $t = t' > 0$, then the general diffusion can go through the locally tree-like structure around this node with the maximum depth,

$$t^* = \lceil t' \log_{a_b} (a_B/a_b) \rceil,$$

for $a_b > 1$, and infinity otherwise. For example, in figure 16 $t^* = \lceil \log_2 4 \rceil = 2$, thus the diffusion could not...
Since \( \partial \) thus the right derivative can be reduced to obtain the right derivative of \( F \) thus so is the function \( F \) derivative in the case of a linear dynamics, where \( \text{Diag} \) is the corresponding diagonal matrix. Specifically, in our case, \( \partial f_t \) is the right derivative, with the right derivative, \( \partial f_t(x) = \begin{cases} 1, & b_t \leq x < B_t, \\ 0, & x < b_t, x \geq B_t, \end{cases} \) so is the function \( F_t \) by the chain rule of semi-differentiability and \( W_{ij} > 0, \forall i,j \). Therefore, we can obtain the right derivative of \( F_t \) with respect to \( x(0) \),

\[
\partial_+ F_t(x(0)) = \partial_+ f_t(y(t))^T \frac{\partial y(t)}{\partial x(t-1)} \prod_{s=1}^{t-1} \text{Diag}(\partial_+ f_s(y(s))) \frac{\partial y(s)}{\partial x(s-1)},
\]

where \( y(t+1) = W^T x(t) \) is the linear part of the state vector \( x(t+1) \), the \( \partial_+ \) is the right derivative with \( \partial_+ f(x) = \partial f(x_i) \), \( \partial \) is the (bidirectional) derivative with \( \partial y/\partial x = (\partial y_i/\partial x_j) \), where \( x, y \in \mathbb{R}^N \), and \( \text{Diag}(\cdot) \) is the corresponding diagonal matrix. Specifically, in our case, \( \partial y(t)/\partial x(t-1) = W^T, \forall t > 0 \), thus the right derivative can be reduced to

\[
\partial_+ F_t(x(0))^T = \partial_+ f_t(y(t))^T W^T \prod_{s=1}^{t-1} \text{Diag}(\partial_+ f_s(y(s))) W^T.
\]  

Since \( \partial_+ f_t(\cdot) \in \{0,1\} \) by (38), the overall right derivative in (39) is a restricted version of the (bidirectional) derivative in the case of a linear dynamics, where \( \partial f_t(\cdot) = 1, \) and

\[
\partial F_t(x(0))^T = 1^T (W^T)^T.
\]  

The right derivative can be useful if the diffusion is dominated by the linear part, but will almost always be 0 if \( b_t \) is close to \( B_t \). Suppose at a particular time step \( t' \), \( b_{t'} = B_{t'} \), then \( f_{t'}(x) = b_{t'} H(x - b_{t'}) \), where \( H(\cdot) \) is again the Heaviside step function, and

\[
\frac{df_{t'}(x)}{dx} = b_{t'} \frac{dH(x - b_{t'})}{dx} = b_{t'} \delta(x - b_{t'}),
\]

where \( \delta(x) \) is the Dirac delta function with \( \delta(x) = +\infty \) if \( x = 0 \) and 0 otherwise. Therefore, the derivative \( \partial F_{t'}(x_i(0)) \) for each node \( i \) and all steps \( t \geq t' \) can only be 0 or \(+\infty\), which is not informative. We conclude here that the derivative information generally has very limited use in understanding the change of diffusion, and accordingly in the IM problem.
B.3 Numerical experiments

Here, we investigate one particular characteristics of the general diffusion model discussed in section B.1, where the dynamics can proceed like in the linear-threshold extreme in one part of the network while act as in the linear-dynamics extreme in another, leading to a coexistence of the two types of diffusion. We know that regular lattices are efficient for the LT model to diffuse, and a tree-like structure is preferred by the linear dynamics but the LT model cannot proceed extensively. Accordingly, we construct a composite network by connecting these two structures, and use as criterion whether the diffusion starting from one part of the network can proceed to the other part.

Specifically, the composite network is composed of the following: (i) a regular lattice of size $N_o$ and with mean degree $k_o$, (ii) a Erdős Rényi (ER) random graph of the same size and with connecting probability $p_{er} = k_o/N_o$, and (iii) edges randomly placed between the two parts with a small probability $p_o$. Here, we choose $k_o = 4$, $N_o = 25$, $p_o = 0.01$, thus the network is of the same size as the SBM in section 5.1 with $N = 50$; we label the nodes in the regular lattice as 0 to 24 and in the random part as 25 to 49; see figure 17 for one realisation. We again assign uniform weight $\alpha = 0.1$. Therefore, $a_b = 2$ corresponds to the critical lower bounds for the linear-threshold extreme, where the LT model cannot influence the whole lattice given limited budget (e.g. $K \leq 3$) with any $a_b > 2$. Hence, we compare the linear-threshold extreme, $a_b = a_B = 2$, with a general case in our diffusion model, $a_b = 2, a_B = 4$, in terms of the number of active nodes up to time step $t'$,

$$\tilde{N}(t';x(0)) = \sum_{i=1}^{N} I_{\{\sum_{t=0}^{t'} x_i(t) > 0\}}$$

where $I_{\{\cdot\}}$ is the indicator function, $x_i(t)$ is the state value of node $v_i$ at time step $t$, and a node is active up to time step $t'$ if it has ever been active at $t \leq t'$, i.e. $\sum_{t=0}^{t'} x_i(t) > 0$.

We observe that, particularly clear in the composite network in figure 17, both sets can influence the whole network in the general case while only reach certain part of the network in the linear-threshold extreme; see figure 18. The two diffusion processes initially proceed similarly in terms of the number of active nodes, because several steps are required for the higher state values to function in influencing more nodes. After this "initial preparation" phase, the general case gradually reach more nodes and finally the whole network. The two plots in figure 19 are obtained from 1000 samples from the ER model. The results from the node set in the regular lattice still have clear separating behaviour, where the general case is expected to influence the whole network. While for the node set in the ER random part, the general case is only expected to reach part of the network with high variance. However, the expectation is still greater than that of the linear-threshold extreme, and also more than half of the network, thus the diffusion is still expected to proceed to the regular-lattice part. The high variance is due to the randomness in the network, which has immediate effect on the diffusion from the random part.
Figure 18: The number of active nodes in the linear-threshold extreme ($a_b = a_B = 2$) and a general case ($a_b = 2, a_B = 4$) with the initially activated nodes in the regular lattice (left) and the ER random part (right), on the composite network in figure 17.

Figure 19: The number of active nodes in the threshold model extreme ($a_b = a_B = 2$) and a general case ($a_b = 2, a_B = 4$) with the initially activated nodes in the regular lattice (left) and the ER random part (right), on 1000 samples of the composite networks.

B.4 Performance evaluation of the CDS method

As discussed in section B.3, both linear-dynamics-like and linear-threshold-like diffusion can coexist in the composite network with the general diffusion model. It is then interesting to explore the performance of the CDS method for the corresponding IM task, on this particular structure. The composite network we refer to is specifically the one in figure 17 constructed by connecting a regular lattice, of size $N_o = 25$ and mean degree $k_o = 4$, and a ER random graph, of the same size and probability $p_{er} = k_o/N$, by a small probability $p_o = 0.01$.

As in section 5.2.2, we still start from changing the upper and lower bound thresholds, $a_b, a_B$, while maintaining the budget size $K = 4$, but directly comparing the performance of the CDS method with and without the community restart strategy. Generally, the plain CDS method has varying performance, where it can find a global optimal solution when the lower bound threshold $a_b$ is small or when the upper bound threshold $a_B$ is high, while the worst-case scenarios come from the region including the linear-threshold extreme; see figure 20. On the one hand, however, if we further investigate the rank, the highest value is approximately 0.01%, hence the CDS method can still find a top 25 solution over all 230,300 sets. On the other hand, the community restart strategy can further improve the performance of the CDS method, where it can now find a global optimal solution in all combinations of the bound thresholds; see figure 21. For those improvements, almost all are from the split close to the inverse ratio of the two communities’ densities.

We further analyse the performance of the CDS method, together with the community restart strategy, when increasing the budget size $K$ while maintaining the lower bound and upper bound thresholds. We
consider the same combinations as in the Karate club network: (i) \(a_b = 2 < a_B = 8\), and (ii) \(a_b = 2 = a_B\).

For case (i) when \(a_b < a_B\), the CDS method has close-to-optimal performance when \(K \geq 3\), with the accuracy greater than 0.98 and the rank decreasing linearly-like in the log-linear plot; see figure 22. When \(K = 2\), even though the accuracy is around 0.8, the rank is still as low as 0.2% in overall 1225 candidates, thus the CDS method can still find a top 3 set. For case (ii) when \(a_b = a_B\), the CDS method performs relatively worse compared to case (i), and this drop in performance is mostly mitigated by the community restart strategy, particularly when \(K = 4\); see figure 23. Note the largest \(K\) we choose here is less than that in the Karate club network, since the time consumption of the brute-force already reaches 2 hours when \(K = 6\), although it only takes a second for the CDS method with the community restart strategy.

### C Time complexity of customised direct search method

In this section, we explore the time complexity of the CDS method empirically on the random ER graphs. Specifically, we construct ER graphs of the same expected node degree, \(< d > = (N - 1)p\) where \(N\) is the network size and \(p\) is the connecting probability in ER graphs, but of increasing size \(N\). We then apply the CDS method to the general IM problem on this series of graphs to explore the dependence of its time complexity on the network size \(N\). The experiments are performed with different combinations of parameters in order to take into account their effects, including the upper bounds \(\{B_t\}\), the lower bounds \(\{b_t\}\), and the number of initially activated nodes \(K\). Here, we assign uniform weight \(\alpha = 0.05\) to all the networks, and maintain the threshold-type bounds in (12) for the general diffusion model, therefore vary the upper and lower bounds through the thresholds \(a_B\) and \(a_b\), respectively.
Overall, the time increases sublinearly while increasing the network size $N$ in ER graphs; see figures 24 for $K = 3$ and 25 for $K = 5$, with different combinations of parameters. When the upper bounds are low, e.g. $a_B = 3$, having a higher expected node degree will not necessarily complicate the problem, since the extra edges may not make a difference in the diffusion and further the search process. When the upper bounds are high, e.g. $a_B = 9$, then the general diffusion model is towards the linear-dynamics extreme and the time consumption is close to that from evaluating the objectives of a full neighbourhood (of size $K(N - K)$). We verify empirically that the diffusion model effectively reaches the linear-dynamics extreme when $a_b = 1$ and $a_B = 9$, since the CDS method stops in the first step and the initial solution is the output. When decreasing the upper bounds or increasing the lower bounds to over $K/2$, the time consumption decreases to some extent, and the decrease is relatively more when the lower bounds are smaller or the upper bounds are higher, respectively. In both cases, the changes are not substantial, since the time maintains the same magnitude, and the trends are consistent with the change in the complexity of evaluating the objective function. Interestingly, when the lower bound threshold $a_b$ increment from 1 to 2 and $K = 5$, the time approximately doubles accordingly. This is because more steps are needed for convergence.

In the ER graphs, $\mathbb{E}[|E|] = N < d > /2$, thus $\mathbb{E}[|E|]$ increases linearly in the network size $N$. From theorem 4.1, the evaluation of the objective function increases (at least) linearly as the number of edges $|E|$, thus here $N$, rises. The size of the feasible discrete neighbourhood, i.e. $K(N - K)$, increases linearly in $N$, since $K = O(1)$. While, from the experiments, we overall observe a sublinear increase in the time consumption with respect to the number of nodes $N$. Therefore, it is reasonable to conjecture that (iib) in section 4.4, the number of evaluations required by the CDS method divided by the feasible neighbourhood size, is at most $O(1)$, in the sense that it will evaluate no greater than $O(1) \times K(N - K)$ candidates until convergence, in the general IM problem in real networks.

The time complexity of the CDS method can be further improved by parallel programming, although
Figure 24: Time consumption of the CDS method on ER graphs of increasing size (x-axis), with $K = 3$ and different combinations of parameters $(a_b, a_B)$, from networks of mean degree 5 (blue) and 10 (orange).

It is described in algorithm 3 and currently performed serially. For instance, the tasks to evaluate the objective function of different candidates in discrete neighbourhood can be run in parallel. There are also other optimisation techniques to further reduce the complexity, e.g. coarse evaluation of the objective function with a higher tolerance at early stage, which can possibly be included in the future work.
Figure 25: Time consumption of the CDS method on ER graphs of increasing size (x-axis), with $K = 5$ and different combinations of parameters ($a_b, a_B$), from networks of mean degree 5 (blue) and 10 (orange).