SIMiner: A Stigmergy-based Model for Mining Influential Nodes in Dynamic Social Networks

Weihua Li, Quan Bai and Minjie Zhang

Abstract—With the widespread of the Internet, the on-line social network with big data is rapidly developing over time. Many enterprises attempt to develop their business by utilizing the power of on-line social networking platforms. A considerable amount of work has focused on how to select a set of influential users to maximize a kind of positive influence in static social networks. However, networks evolve, and the topological structure changes over time. How to mine and adapt the influencers in a dynamic and large-scale environment becomes a challenging issue. In this paper, a collective intelligence model, i.e., stigmergy-based influencers miner, is proposed to investigate influential nodes in a fully dynamic environment. The proposed model is capable of analysing influential relationships in a social network in decentralized manners and identifying the influencers more efficiently than traditional seed selection algorithms. Moreover, it is capable of adapting the solutions in complex dynamic environments without any interruptions or recalculations. Experimental results show that the proposed model achieves better performance than other traditional models in both static and dynamic social networks by considering both efficiency and effectiveness.

Index Terms—Ant Algorithm, Stigmergy, Multi-Agent System, Influence Maximization

1 Introduction

Over the last decade, with the development and prevalence of on-line social networks, the public opinions with big data emerge and disseminate rapidly [1]. By relying on such features, viral marketing becomes one of the typical sales strategies, which aims to spread the positive influence of a specific product or brand through 'word-of-mouth' effects [2]. This business strategy is capable of increasing brand awareness and achieving marketing objectives effectively and efficiently. One significant task in viral marketing is influence maximization [3], [4]. It aims to select a limited number of influential nodes, which can propagate influence as much as possible in a network. The selected group of influencers is called *seed set*, and the seeding process is named as *seed selection*. Influence maximization is NP-hard, and approximation approaches are normally considered as a replacement.

The major challenging issues of influence maximization problem stem from two intrinsic characteristics of the social networks, i.e., large scale and dynamics, and both often come up together in reality. Most social networks possess a vast number of nodes and links, as well as evolving topological structures. Nodes keep growing; links are formed and vanished; link strengths revise over time. To handle the dynamics of influence maximization, when the network topological structure changes, traditional seed selection approaches are supposed to recalibrate the solutions by running over again, which is computationally expensive. Moreover, using numerous static snapshots of a social network is a typical representation of dynamics but becomes unrealistic in a large-scale and continuously evolving environment. It is obvious that capturing snapshots in seconds in a big-data environment will inevitably create another set of big data, making the existing problem more complicated. On the other side, almost all the diffusion models require a global view of the social network for influence-diffusion simulations. However, in many applications, local information is available merely, e.g., the practitioners intervene in a population, where the observation of the social network is initially not discovered, and the data is required to be collected via a laborious process [5], [6]. Therefore, by considering both features, it is extremely difficult to analyse and mine influential users by leveraging classic diffusion models and seeding algorithms.

The collective intelligence approach strongly contributes to the shift of knowledge from individual to the collective, which appears more competent in handling the large-scale, dynamic and distributed environment. It disrupts the limitations and handles dynamics at the microscopic level using agents. Specifically, individuals revert the latest information by exploring the possible solution in a dynamic context through communications; thus, the solution can be adapted over time based on the evolutionary environment. Furthermore, the decentralized model cuts the computational cost by sharing the workload and distributing the tasks to individuals, so as can be operated without a global view. Therefore, such approaches appear applicable in many applications domains with undiscovered global views, e.g., WeChat business [7]. There are basically two major types of decentralized models regarding communications. One relies on direct communications among the individuals, such as cellular automata [8], where each cell in the grid adapts its state by looking at the adjacent neighbours based on a set of rules. The other focuses on indirect communications by reading or analysing the messages left by the peers. Stigmergy-based models [9] are a typical type of decentralized models applicable to the second category.

W. Li and Q. Bai are with the School of Engineering, Computer & Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand, E-mail: weihua.li@aut.ac.nz; quan.bai@aut.ac.nz
M. Zhang is with the School of Computing and Information Technology, University of Wollongong, NSW, Australia, E-mail: minjie@uow.edu.au

Stigmergy is defined as “stimulation of workers by the performance they have achieved” [10], which is a particular indirect communication mechanism exhibited by tiny social insects, such as ants, to coordinate group activities. Their indirect communications are normally conducted through leaving a chemical substance, i.e., pheromones, on the trails, which evaporate over time. Inspired by the stigmergic interactions, the stigmergy and ant algorithms have been widely applied in many applications without global information, such as communication network routing, exploratory data analysis and diagram drawing, where the intelligent agents cooperate with each other by leaving and sensing the artificial pheromone, which indicates application-specific information [11]. In [12], we introduced some preliminary results from investigating the feasibility of stigmergy-based approach applied in the influence maximization problem. To the best of our knowledge, it is the first literature introducing the application of stigmergic interactions in this field. However, the proposed approach in [12] neglects the temporal features and dynamics of a social network. The adapting capabilities of handling a changing environment are neither modelled or systematically articulated. Moreover, the model only considers the limited information, e.g., the number of the neighbours, when allocating the pheromone. This potentially results in bias in terms of influence.

In this paper, we propose a collective intelligence model called Stigmergy-based Influencers Miner (SIMiner), which is able to analyse and mine influential nodes in dynamic social networks in a decentralized manner. The proposed model is applicable in both static and dynamic complex environments, and capable of adapting the solutions in an evolving context. SIMiner is a Multi-Agent System (MAS) [13], which incorporates two types of agents, i.e., user agents and ant agents. The former provide information actively for ant agents when requested. While the latter traverse the network based on the heuristics, seeking for the influencers in a social network. Ant agents’ key behaviours, including tour formation and pheromone allocation, have been modelled for selecting appropriate influential nodes to achieve the maximum positive impact. The pheromone evaporation, another factor affecting the pheromone deposition, is also formulated in SIMiner. Specifically, tour formation refers to how ants walk and form a tour in the environment, and the latter aims to distribute pheromone to specific nodes based on the results of a local influence diffusion model. While pheromone evaporation is an exploration mechanism that delays faster convergence of the solutions. Empirical experiments have been conducted to analyse the convergence and evaluate the performance of SIMiner against other classic models under both static and dynamic social networks. The experimental results reveal that the proposed model can converge to an optimal solution gradually and function perfectly in a distributed environment without a global view. It outperforms state-of-the-art approaches by considering both influence efficiency and effectiveness. Moreover, SIMiner is able to mine influencers in dynamic environments without any interruption or recalculation, and the solution can be adapted quickly over time. To summarize, the contributions of this research work are as follows.

- We first systematically articulated the influence maximization using stigmergy approach in a distributed environment. In other words, to the best of our knowledge, SIMiner is the first distributed model in this field.
- We leveraged SIMiner to tackle the challenging issues, i.e., the large-scale and dynamic environment, from a microscopic level using multi-agents.
- We explored the convergence of SIMiner, as well as the impact of varying parameters, showing that the efficiency of the mining capability can be improved by simply adding the same ant agents to the environment.
- We analysed and evaluated the performance of SIMiner. The empirical results reveal that SIMiner can give excellent performance, and even better than that of greedy algorithm in some datasets. Greedy algorithm in the influence maximization problem outperforms most of the proposed algorithms but is not scalable.

The rest of this paper is organized as follows. Section 2 reviews the literature related to this research work. Section 3 introduces the preliminaries and gives formal definitions. Section 4 systematically elaborates SIMiner model. Theoretical analysis of convergence is conducted in Section 5. In Section 6, experiments and experimental results are presented to evaluate the performance of SIMiner. Our conclusions are detailed in Section 7.

2 Related Work

2.1 Influence Maximization

In on-line marketing, it is very important to investigate how to propagate positive influence in a social network with limited resources. Motivated by this background, Kempe et al. formulate the influence maximization as a discrete optimization problem, which aims to select a limited set of influential users from the network, expecting that they can effectively spread the positive influence across the entire network [4]. Several popular seed selection approaches are presented in the contemporary literature, such as greedy selection, degree ranking selection and random selection, where greedy selection outperforms most of the existing algorithms but appears not scalable.

Greedy algorithm attempts to reach the maximum influence marginal gain in selecting each seed, coming with a $(1 - 1/e)$ approximation guarantee. This results from properties of monotonicity and sub-modularity that the spread function exhibits under some diffusion models [4]. The greedy algorithm facilitated in the influence maximization problem is simulation-based, which selects each seed by running numerous times of Monte-Carlo simulations. For example, in seed selection, the first seed can be identified by estimating each individual’s influence through the simulations. Next, having the first influencer involved, the second seed can be identified by estimating the influence coverage of all the possible combinations, and so on and so forth. Eventually, the seed set will be selected.

Many studies have been conducted to improve the efficiency and effectiveness of seed selection algorithms. Chen et al. study the efficient influence maximization by improving the original greedy selection, and propose a novel seed selection approach, namely, degree discount heuristics for the uniform Independent Cascade model [2]. Goyal et al. propose a novel CELF algorithm, i.e., CELF++, to reduce the running time [14]. Zhang et al. investigate the Least Cost Influence problem (LCI) in multiplex networks, and the LCI problem is alleviated by mapping a set of networks into a single one via lossless and lossy coupling schemes [15].

However, all these approaches can neither handle the dynamics of social networks nor function without a global view.

2.2 Dynamic Influence Maximization

How to handle influence propagation in dynamically temporal social networks also has drawn attention to some researchers [16], [17], [18]. Zhuang et al. study influence maximization in dynamic social networks and first introduce the concept of probing in dynamic networks. Similarly, Bao et al. study influence maximization in dynamic social networks by proposing an on-line randomized algorithm dealing with both unknown and non-stationary influence probabilities [19]. In [20] and [21], traditional Independent Cascade (IC) model and Linear Threshold (LT) model have been extended to capture the dynamics aspect of real social networks. Karim and Holme attempt to handle the dynamics of users' interactions by proposing a threshold model of cascades [22]. Gomez-Rodriguez et al. develop a flexible model, i.e., NetRate of the spatio-temporal structure underlying diffusion process [23]. Whereas, these approaches consider only a few dynamic features of a social network, such as dynamic propagation probabilities. The fully dynamic topological structure of real social networks is not handled when modelling the influence diffusion. With an exception, Naoto et al. propose the first real-time fully-dynamic index data structure designed for influence analysis on evolving networks, where five network operations: vertex additions, vertex deletions, edge additions, edge deletions, and propagation probability updates are included [24]. However, the model in [24] appears centralized and requires a global view. Furthermore, it needs to update the index according to the graph changes.

By contrast, the collective intelligence approach proposed in this paper only concentrates on modelling the features and behaviours of agents and can handle the topologically and temporally dynamic network automatically. Thus, the proposed model possesses excellent adaptation capability to explore solutions in a changing environment.

2.3 Ant and Stigmergy Algorithm

Ant and stigmergy-based algorithms do not necessarily require global network information, and the computation is decentralized. Stigmergy relies on the ant colony knowledge, as it demonstrates a particular mechanism exploited for indirect communication among ants to control and coordinate their tasks. In natural environments, stigmergy-based systems have shown that they can be utilized for generating complicated and robust behaviours in the systems even if each ant has limited intelligence. Some researchers have applied stigmergy in the computer science field. Dorigo et al. introduce how to solve the Travelling Salesman Problem (TSP) [25] by leveraging ant and stigmergy-based algorithms, where the pheromones are allocated by considering the distances among the cities [9]. Ahmed et al. propose a stigmergy-based approach for modelling dynamic interactions among Web service agents in decentralized environments [11]. Takahashi et al. propose an anticipatory stigmergy model with allocation strategies for sharing near future traffic information related to traffic congestion management in a decentralized environment [26]. Hadeli et al. introduce a novel design and prototype implementation for manufacturing control systems using stigmergy, which tends to handle the changes and disturbances [27]. Lewis claims that the essential social networking behaviours of human beings are in fact forms of stigmergy, and attempts to explain a theory of group formation based on stigmergy [28].

However, the stigmergy-based algorithm is not fully utilized in the influence maximization problem though it demonstrates its superior in handling optimization problems in a distributed manner.

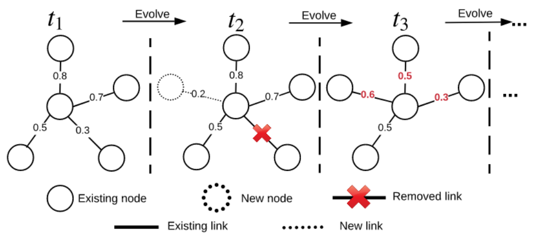

Fig. 1. Fully Dynamic Social Network

3 Preliminaries and Formal Definitions

3.1 Fully Dynamics of Social Networks

Modelling emergent properties of social networks appears to be one of the pillars of social network science [16]. Real-world social networks possess a highly dynamic nature and evolve rapidly over time [29], [30]. More importantly, the network evolution is continuous. Near all of the approaches in this research field utilise numerous static snapshots of consecutive discrete time steps to mimic the dynamics of social networks. For example, a fully dynamic social network can be represented in Figure 1. As shown in Figure 1, a social network evolves and updates over time, i.e., from time steps t_1 to t_n, incorporating the addition and deletion of nodes and links, as well as the weight updates.

However, for large-scale networks, it is almost impossible to imitate dynamics by capturing snapshots or storing all changes happening around since this inevitably creates another set of big data. Furthermore, capturing snapshots or changes require a global view, which becomes another obstacle as a central component is required for monitoring the entire network in real-time.

To overcome the difficulties mentioned above, we argue that the stigmergy-based multi-agent system is more suitable for capturing the dynamic behaviours of social networks since individual agents have been deployed to explore the real-time changes in a local level from a microscopic point of view. This approach functions even without a global view.

3.2 Stigmergy and Multi-Agent Systems

In general, stigmergy refers to a series of behaviours initiated by simple animals, such as ants, termites and wasps, which is a collection of mechanisms that mediate the interactions among these animals [28], [31]. The communications among the individuals are mediated by a sort of biological substance, i.e., pheromone. By borrowing the ideas from this biological phenomenon, many researchers model the ants as autonomous and self-directed agents [9], [11], [26]. All of the agents work in the same context and form a MAS. The agents exchange messages indirectly. Specifically, they keep modifying the global environment by leaving certain amounts of pheromone trails based on their local experiences and discoveries. In the meanwhile, the agents select walking paths according to the pheromone concentration and a partial network topological structure covered in the local view at a particular time frame. Therefore, the pheromone concentration and distribution represent the problem that the agents are currently working on [27]. From a microscopic perspective, the MAS demonstrates an evolutionary pattern driven by the local behaviours of individual agents.

In this paper, we model the social network as the dynamic context/environment of SIMiner, where two types of agents reside in the same context, i.e., *user agents* and *ant agents*. The former represent users or nodes in a social network, while the latter refer to the agents for analysing users' influential degrees. Each user agent possesses a local view covering the neighbourhood. Similarly, ant agents can move among user agents and sense the environment at a local level. Both types of agents are capable of interacting with each other, while ant agents can only communicate with their species indirectly. On the other side, the influence propagation process is simulated as the crawling behaviours of ant agents. The objective of the ant agents is to investigate the possible influential users from the social network. To explain this further, the ant agents keep moving through the network and allocate different amounts of pheromone on each node that they claw over as rewards. The pheromone rewarding strategy is on the basis of a local influence propagation model. The users with higher influential abilities will gain more pheromone and become prominent after a number of iterations.

In SIMiner, multiple ant agents keep crawling in a dynamic context simultaneously and iteratively. Figure 2 illustrates an overall picture of SIMiner, implying how SIMiner handles the dynamics from a microscopic level and in a distributed environment. Three ants are presented in this figure. Each ant captures the environment within the coverage of its local view when it reaches any node at a particular time step. Any environmental updates prior to the arriving of ants are not supposed to be discovered, but the modifications will be spotted when they are within the local view of any ants. From a microscopic viewpoint, ants do not care about the changes of network topological structures; instead, they concentrate on local time-spatial observations. Therefore, SIMiner neither replies on a global view nor proactively captures global network snapshots but is capable of handling the dynamics locally, which explicitly simplifies the problem significantly.

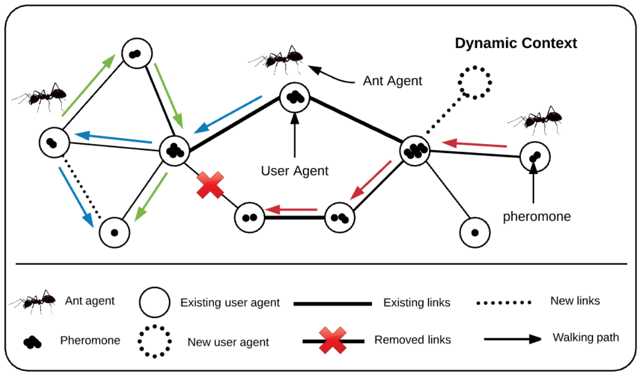

Fig. 2. Stigmergy-based Influencers Miner Model

3.3 Influence Diffusion Models

Most researchers have conducted social influence analysis and modelling based on two fundamental influence diffusion models, i.e., IC model and LT model, which have been widely applied in influence maximization problem [4]. In both models, each node has two states: an active state and an inactive state. In the beginning, a limited set of users, i.e., seed nodes, are supposed to be selected as the initial active nodes, which attempt to propagate influence and affect the inactive neighbours at a certain chance. If any neighbour is activated, the state of the node becomes active, and it starts to propagate influence to the neighbours. Moreover, both models have two key properties, i.e., propagation and attenuation. The influence is initiated from the seed set, i.e., activated nodes. These nodes transfer their influence through the correlation graph, whereas the power of this effect decreases when hopping further and further away from the activated nodes. IC model and LT model represent different deterministic strategies of influence though they share some common features.

IC model is a kind of non-deterministic diffusion model, where the receiver's state is not deterministically decided by itself but is affected and influenced with a predefined probability by the senders [32]. In IC model, when an active node v_i interacts with the adjacent inactive nodes $\Gamma(v_i)$, it has a single chance to activate each neighbour at a successful rate, whereas, the features of nodes are not taken into consideration [33]. In contrast, LT model is a deterministic diffusion model [32]. In this model, each node is assigned a fixed threshold, where the threshold of node v_i can be represented as ϑ_i. Node v_i is influenced by the neighbours $\Gamma(v_i)$ when the sum of their weights exceeds ϑ_i. The threshold controls the opinion or state adoption for each node. Specifically, nodes with a high threshold have low probabilities to be influenced, while for those low-threshold nodes, they have a high tendency of being active.

Both IC model and LT model can be facilitated to simulate the diffusion process when given a static network with a global topological structure. However, they cannot function only with local views; both require a substantial extension to handle the temporal features of a social network [20], [21], [22]. Moreover, the outcome of both traditional models merely demonstrates a global activation coverage and ignores the contribution of each individual.

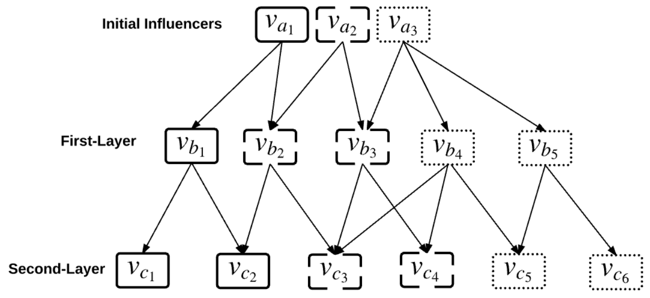

Fig. 3. An Independent Cascade Model with Pyramid Scheme

In SIMiner, we propose and leverage an extended version of the IC model, i.e., Independent Cascade Model with Pyramid Scheme (ICMPS), to measure the individual's activation contribution or influential capabilities in a local social network, which provides the evidence for pheromone distribution. ICMPS inherits the key features from the IC model, where influence diffusion is demonstrated as a hopping and infecting process. Whereas ICMPS tends to capture the activation contribution of the influencers in a local environment, and the activation follows a pyramid pattern. Figure 3 illustrates a toy example of ICMPS in a static local network, where the three initially active users, i.e., v_{a_1}, v_{a_2} and v_{a_3}, initiate the influence effecting on the two-hop neighbourhood. As we can observe from the figure that v_{a_1} successfully activates its neighbour v_{b_1} only; both v_{b_2} and v_{b_3} are activated by v_{a_2}; v_{a_3} influences v_{b_4} and v_{b_5}. Next, the newly active nodes in the first layer attempt to exert influence on their direct neighbours, i.e., the nodes in the second layer. Therefore, activation contribution of v_{a_1}, v_{a_2} and v_{a_3} is 3 ,4 and 4, respectively. As ICMPS is a stochastic model, the results are supposed to be averaged over multiple trials.

3.4 Formal Definitions

In general, a social network at a particular time step k, i.e., $G(k) = (V(k), E(k))$, can be defined as a graph containing numerous of entities $V(k)$ with their connectivities $E(k)$, and it possesses an evolving and dynamic topological structure. In this research, we model the problem space as a distributed MAS. The environment at time step k, $Env(k) = (V(k), A)$, is considered as a shared working space of two types of agents, i.e., user agents $V(k) = \{v_1, v_2, ..., v_j\}$ and ant agents $A = \{a_1, a_2, ..., a_i\}$. In static networks, $\forall k \in \mathbb{N}$, $G(k) = G(k+1)$ and $Env(k) = Env(k+1)$. For simplification purpose, "(k)" can be omitted in a general context.

Definition 1: A user agent (node) v_i refers to an agent in the environment Env. $v_i.E$ denotes the friendship set/attribute of user v_i, where $v_i.E = \{e_{ij} | v_j \in V \land e_{ij} \in v_j.E, v_i \neq v_j\}$. A particular element in $v_i.E$ can be represented as a three-tuple, i.e., $e_{ij} = (v_i, v_j, w_{ij})$, where w_{ij} is the weight of e_{ij}, describing the affiliation strength, and it also can be denoted by using the notation $e_{ij}.w$. Meanwhile, any user agent v_i has a set of neighbours, i.e., $\Gamma(v_i) = \{v_j | e_{ij} \in v_i.E \cap v_j.E, v_i \neq v_j\}$. The friendship affiliation information is supposed to be maintained by each individual locally.

Definition 2: An ant agent a_m is defined as an autonomous agent working for a specific problem in the environment Env, which crawls across the nodes in the same environment based on users' relationships. The friendship affiliation signifies the available routes for ant agents.

There exist a number of ant agents in the environment, $A = \{a_1, a_2, ..., a_i\}$. Moreover, they are capable of communicating with user agents in order to discover and evaluate the amount of pheromone on the current user and the ones nearby, as well as examine the strength of the relationships.

Definition 3: A tour $T_m^n = (\overrightarrow{V_m^n}, \overrightarrow{\tau_m^n})$ is defined as the path that ant agent a_m walks through d nodes in Round n, where $\overrightarrow{V_m^n} =< v_1, v_2, ..., v_d >$ denotes a directed vector which contains the node sequence in the tour, while $\overrightarrow{\tau_m^n} =< t_1, t_2, ..., t_d >$ refers to the corresponding time when a_m passing each element in $\overrightarrow{V_m^n}$. For simplification purpose, we regard T as a tour in a general context.

Specifically, a_m randomly selects a starting point (a user agent). Next, it crawls from one node to currently existing adjacent neighbours at the next time step and eventually ceases when it reaches the *endpoint* v_d, where $\Gamma(v_d) \subset \{v_1, v_2, ..., v_d\}$.

Definition 4: N-layer Sub-network $G_m^n(N) = (V_m^n(N), E_m^n(N))$ is defined as a local static sub-graph generated by ant a_m after completing tour T_m^n. The edge set $E_m^n(N)$ includes all the links among $V_m^n(N)$, while $V_m^n(N)$ denotes the union set of the nodes in T_m^n and the corresponding N-hop neighbourhood. Thus, $V_m^n(N)$ can be formulated in Equation 1.

$$V_m^n(N) = \begin{cases} \{v_1, v_2, ..., v_d\}, & N = 0 \\ V_m^n(N-1) \cup \Gamma(V_m^n(N-1)), & N \neq 0 \end{cases} \tag{1}$$

Definition 5: Pheromone represents the information and heuristics passed by ant agents to their peers based on the local experiences. $p_m^n(v_i)$ represents the pheromone amount allocated to v_i in tour T_m^n. While $v_i.p(n)$ indicates the pheromone amount accumulated on user agent v_i in the n^{th} interactive walking around, and the value is constantly changing over time.

Pheromone also can be regarded as the reward granted to user agents. Specifically, each ant agent tends to allocate more pheromone on the nodes which exert high impact potentially. In other words, the more pheromone amount a user agent (node) possesses, the higher chance it becomes an influencer.

4 Stigmergy-based Influencers Miner Modelling

SIMiner inherits several major features of the ant algorithm and agent-based modelling, which is capable of dynamically mining influencers by adapting the solutions over time. In SIMiner, numerous ant agents crawl simultaneously and update the shared environment by allocating pheromones on user agents. The influence propagation process is simulated as crawling behaviours of ant agents. The influencers

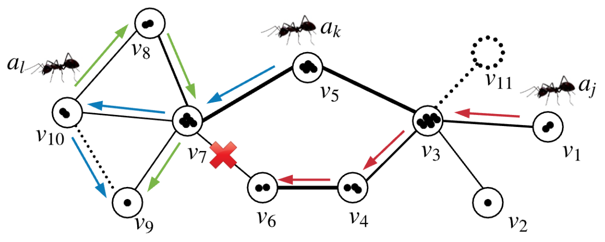

Fig. 4. An Annotated Dynamic Social Network

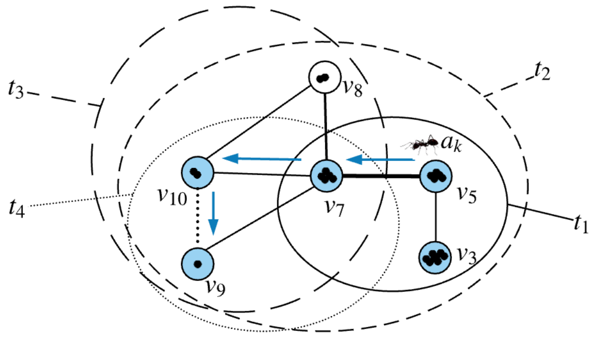

Fig. 5. Dynamic Local View of Ant a_k based on Different Time Steps

can be selected based on the pheromone distribution and concentration in the network. The detailed modelling of SIMiner is elaborated in the following subsections.

4.1 Overall Process of SIMiner

In this subsection, we explain the overall process of SIMiner, including **(1)** how to handle the dynamic temporal and topological features of a social network, and **(2)** how to mine influencers using SIMiner.

First, the network is converted into a distributed environment shared by two types of agents, i.e., user agents and ant agents. By using the same sample dynamic network in Figure 2, we annotate all the agents and demonstrate in Figure 4. Assume that all the ant agents start tours at the same step t_1, and it takes one time step to crawl from one node to another. As we can observe that all of the three ants complete the tour after three hops. Each ant agent possesses a dynamic local view which is driven by the changing structure of a social network, both temporally and spatially. Figure 5 shows the dynamic local view of ant a_k ranging from t_1 to t_4. It is obvious that each ant agent only focuses on the local context at a particular time step.

In Figure 4, we assume the following three events and corresponding time steps. Thus, the dynamic local view of ant agents at each time step is described in Table 1.

- Event 1: Link $e_{9,10}$ is formed at t_2. v_9 is not covered in ant a_l's local view at t_1, but included at t_3 and t_4.
- Event 2: Node v_{11} joins at t_3. v_8 is not spotted by a_j when it passes v_3 at t_2.
- Event 3: Link $e_{6,7}$ vanishes at t_3. Ant a_k can observe v_6 at t_2, but a_l cannot see v_6 when it reaches v_7 at t_3. Similarly, a_j arrives the end of its tour at t_4, due to the broken linkage.

Second, ant agents embark on the exploration of influencers concurrently, and they collaborate with each other by distributing pheromone on the nodes. The behaviours of an ant agent incorporate selecting paths, constructing local influence propagation sub-networks and distributing pheromones. Meanwhile, pheromone evaporation happens over time, which enables SIMiner to forget the poor choices in the past. When solutions start to converge, i.e., the pheromone ranking list has few changes, the influencers can be easily located purely based on the pheromone amount.

4.2 Tour Formation

Tour formation ascribes to a series of path selection and skipping behaviours from an ant agent.

4.2.1 *Path Selection*

Path selection is one of the ant's fundamental behaviours. An ant agent a_m needs to select the next node to crawl when having various choices $V_c = \{v_j | v_j \in \Gamma(v_i)\}$, where v_i denotes the node that a_m arrives at.

Basically, the path selection decision is based on two major aspects, incorporating the pheromone amount of v_j, i.e., $v_j.p$, and the weight of corresponding edge $e_{ij}.w$. The path selection behaviour is modelled as a probabilistic event by using Equation 2, where q_{ij} denotes the probability that an ant agent walks from v_i to v_j, α and β are two parameters balancing the weight of two factors.

$$q_{ij} = \begin{cases} \frac{(e_{ij}.w)^{\alpha} \cdot (v_j.p)^{\beta}}{\sum\limits_{v_x \in \Gamma(v_i) \setminus T} (e_{ix}.w)^{\alpha} \cdot (v_x.p)^{\beta}}, & e_{ij} \in v_i.E \\ 0, & e_{ij} \notin v_i.E \end{cases} \tag{2}$$

Furthermore, ant agents cannot choose the nodes they have walked through within the same tour, but can pass the nodes that other ant agents have walked before the current or previous round.

4.2.2 *Skip*

Walking from v_i to v_j, an ant agent can either select v_j or skip it by comparing the ratio of common neighbours between v_i and v_j. If the ratio is greater than a predefined threshold, v_j is supposed to be skipped. Subsequently, node v_j will be neither selected into the tour nor considered for pheromone allocation.

The rationale of defining "skip" behaviour can be reflected in two folds. **(1)** "Skip" is initiated based on the fact that many of the most central nodes in a social network may be clustered; thus, it is not necessary to target all of them [2], [34]. For example, if two users, v_i and v_j, share many common neighbours, both are most likely influenced by each other since they own a broad range of communicational channels. The same concept has been applied in the traditional influence maximization problem. In other words, if one is selected as a seed, the other should be ignored, since selecting both cannot enlarge the global activation coverage to a large extent, and the impact generated by both may be pretty much close to choosing either of them. **(2)** "Skip" helps to tailor the ant algorithms to the influence maximization problem. Different from TSP, the identified influencers are not necessarily connected with each other or

TABLE 1
Ant Agents' Local View Overtime

	t_1	t_2	t_3	t_4
a_j	v_1, v_3	v_3, v_5, v_4, v_1, v_2	v_4, v_6, v_3	v_6, v_4
a_k	v_5, v_7, v_3	$v_7, v_8, v_{10}, v_9, v_5$	v_{10}, v_8, v_7, v_9	v_9, v_{10}, v_7
a_l	v_{10}, v_8	v_8, v_{10}, v_7	$v_7, v_8, v_{10}, v_9, v_6, v_5$	v_9, v_{10}, v_7

stay in the same path, but they can be scattered everywhere in the entire social network. "Skip" chops a continuous path selected by an ant agent into pieces, which satisfies the pattern of the solutions for influence maximization. This behaviour significantly improves the quality of the selected influencers, which is explained in the Experiment section.

In SIMiner, ant agent a_m skips v_j if the common neighbour percentage comparing with previous node v_i, i.e., η_{ij}, exceeds a certain limit ω, where η_{ij} can be measured by using Equation 3.

$$\eta_{ij} = \frac{|\Gamma(v_i) \cap \Gamma(v_j)|}{|\Gamma(v_j)|} \tag{3}$$

4.2.3 Tour Formation Algorithm

As introduced in Definition 3, a tour can be represented as a sequential list of nodes and the corresponding time step of passing each node. In the shared environment, each ant agent keeps performing an iterative process: walking and selecting paths. Whereas, the actions stop when the ant agent reaches the endpoint v_d. In other words, the iterative process triggered by ant agent a_m in Round n produces a path vector, i.e., tour T_m^n.

Algorithm 1 Tour Formation Algorithm

```
Input: $a_m, n, t_0$
Output: $T_m^n = (\overrightarrow{V_m^n}, \overrightarrow{\tau_m^n})$
1: Initialize $a_m$ and randomly select a starting point $v_s, v_s \in V$
2: Initialize a tour vector $\overrightarrow{V_m^n} :=<>$, a tour set $V_m^n := \emptyset$
3: while $\Gamma(v_s) \not\subset V_m^n$ do
4:     Initialize candidate list $V_c := \emptyset$
5:     for $\forall v_i \in \Gamma(v_s) \wedge v_i \notin V_m^n$ do
6:         Compute the probability $q_{si}$ using Equation 2.
7:         if $q_{si} > \sigma$ then
8:             $V_c := V_c \cup \{v_i\}$
9:         end if
10:    end for
11:    $v_s := null$
12:    if $V_c \neq \emptyset$ then
13:        Choose the next node $v_n \in V_c$
14:        Compute $\eta_{sn}$ using Equation 3
15:        if $\eta_{sn} \leq \omega$ then
16:            $\overrightarrow{V_m^n} :=< \overrightarrow{V_m^n}, v_n >$
17:            $\overrightarrow{\tau_m^n} :=< \overrightarrow{\tau_m^n}, t_0 >$
18:            $V_m^n := V_m^n \cup \{v_n\}$
19:        end if
20:        $v_s := v_n$, $t_0 := t_0 + 1$
21:    end if
22: end while
```

Algorithm 1 describes the tour formation process. The inputs of this algorithm include ant agent a_m, round index n and the starting time step t_0, while the output is tour T_m^n. Line 3 shows the criteria of walking to the next node. Lines 5-10 demonstrate the targeting candidates selection, where σ is a predefined threshold to filter out those candidates with low probability. Lines 11-21 indicate the path selection process. Whereas, Lines 12-15 tend to judge whether an node should be included in the tour. The iterative walking process ends when all of the current node v_s's neighbours reside in tour T_m^n.

The complexity of Algorithm 1 is mainly determined by the loops in Line 3 and Line 5. The complexity is $O(ls)$, where l presents the length of the path and s denotes the cardinality of the neighbourhood.

4.3 Pheromone Operations

Pheromone plays a significant role in the proposed model since it represents information or heuristics based on the ant agents' experiences which can be referred by their peers. In this section, pheromone operations, including evaporation, N-layer sub-network generation and allocation will be introduced in the following subsections.

4.3.1 Pheromone Evaporation

Pheromone evaporation is a common phenomenon, where the amount of allocated pheromone decreases over time. In stigmergy-based algorithms, this mechanism delays the faster convergence and avoids to converge to a locally optimal solution [35]. In the current model, pheromone evaporation increases the randomness and encourages ant agents to select divergent walking paths. Furthermore, pheromone evaporation eliminates the poor choices in the past, so that the present influencers become more prominent.

Pheromone evaporates through each node within the scope of the whole network at the same time. At a justified time, all of the nodes in the network will evaporate a predefined unit of pheromone. The pheromone evaporation is quantified by using Equations 4 and 5, where ρ_n denotes evaporation rate, n represents the iterative time step, and a is a constant.

$$v_j.p(n+1) = v_j.p(n) \cdot \rho_n \tag{4}$$

$$\rho_n = 1 - \sqrt{\frac{n}{n+a}} \tag{5}$$

4.3.2 N-Layer Sub-network Generation

N-layer sub-network generation is the preliminary step of pheromone allocation, which constructs a local environment to estimate the influence contribution of each individual in a tour. "N" is natural number describing the depth of influence propagation.

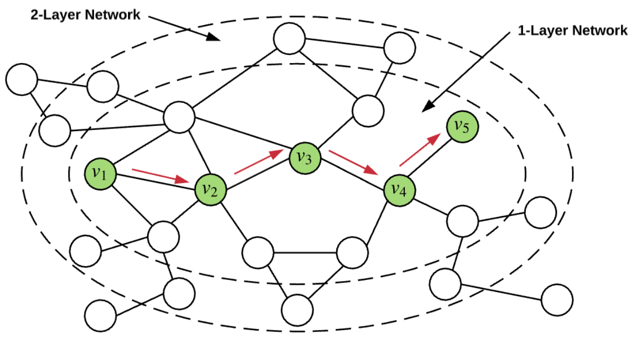

Fig. 6. N-Layer Sub-Network

Once ant agent a_m completes tour T_m^n, a corresponding local N-layer sub-network $G_m^n(N) = (V_m^n(N), E_m^n(N))$ is supposed to be generated based on the path that a_m walked through. $V_m^n(N)$ incorporates all the elements in tour T_m^n and the corresponding N-hop neighbourhood. For example, given $N = 2$, $V_m^n(2) = V_m^n(0) \cup \Gamma(V_m^n(0)) \cup \Gamma(\Gamma(V_m^n(0)))$. While the edge set $E_m^n(2)$ includes all the links among $V_m^n(2)$.

Figure 6 describes an example of a generated two-layer sub-network, i.e., $G_m^n(2)$. Ant a_m completes tour T_m^n in Round n, where $\overrightarrow{V_m^n} =< v_1, v_2, v_3, v_4, v_5 >$. The pheromone distribution initiated by each ant agent relies on the generated N-layer sub-network.

4.3.3 Pheromone Allocation

Pheromone allocation, in general, refers to how ant agents determine the amount of biological information left on the nodes that they have crawled over. From a broader point of view, the pheromone allocation unravels the rewarding strategies initiated by ant agents. Basically, ant agents tend to reward the potential influential users and update the environment by modifying the pheromone amount on each node. Thus, it demonstrates an evolutionary group work, and the solution is continuously being optimized.

The pheromone allocation is determined based on the generated N-layer sub-network; more pheromone will be allocated to the nodes in a sub-network with larger size and a path with a shorter length generally.

Next, the approach to disseminate pheromone to the environment is elaborated as follows. The pheromone amount Δq_m^n distributed by a_m to node $v_i, v_i \in T_m^n$ is proportional to the contribution of v_i in the tour. The individual's activation contribution of v_i in T_m^n, i.e., $c_m^n(v_i)$, refers to the number of nodes that have been activated by v_i in a sub-network, which can be calculated through the ICMPS model. The activation contribution of $v_i, v_i \in V_m^n(N)$, is denoted by using $c_m^n(v_i)$, and $p_m^n(v_i)$ can be formulated in Equation 6.

$$p_m^n(v_i) = g(v_i) \cdot \rho_n \tag{6}$$

In this equation, $g(v_i)$ refers to an adjustment function of v_i formulated in Equation 7, where D denotes a constant. $g(v_i)$ is proportional to $c_m^n(v_i)$.

$$g(v_i) = \frac{c_m^n(v_i)}{c_m^n(v_i) + D} \tag{7}$$

Therefore, by considering both evaporation and allocation, the pheromone update of v_i after tour T_m^n completed is described in Equation 8.

$$v_j.p(n+1) = \begin{cases} v_j.p(n) \cdot (1-\rho_n) + p_m^n(v_i) & ,if\ v_j \in T_m^n \\ v_j.p(n) \cdot (1-\rho_n) & ,otherwise \end{cases} \tag{8}$$

Algorithm 2 Pheromone Allocation Algorithm

```
Input: $T_m^n = (\overrightarrow{V_m^n}, \overrightarrow{\tau_m^n}), N$
Output: $\{v_m.p | v_m \in \overrightarrow{V_m^n}\}$
 1: Initialize $G_m^n(N) = (V_m^n(N), E_m^n(N))$
 2: Initialize $V_m^n(N) := \emptyset, E_m^n(N) := \emptyset$
 3: Initialize temp variable $\kappa = 0$
 4: for $\forall v_i \in \overrightarrow{V_m^n}$ do
 5:     Obtain the next element $v_j$ from $\overrightarrow{V_m^n}$
 6:     Calculate $\eta_{ij}$ using Equation 3.
 7:     if $\eta_{ij} \le \omega$ then
 8:         $V_m^n(N) := V_m^n(N) \cup \{v_i\}$
 9:     end if
10: end for
11: while $\kappa < N$ do
12:     for $\forall v_i \in V_m^n(N)$ do
13:         $E_m^n := E_m^n \cup v_i.E$
14:     end for
15:     $V_m^n(N) := V_m^n(N) \cup \Gamma(V_m^n(N))$
16:     $\kappa := \kappa + 1$
17: end while
18: Initialize seed set $S := \overrightarrow{V_m^n}$
19: for $\forall v_x \in S$ do
20:     $v_x.activatedBy := v_x$
21: end for
22: Given $G_m^n(N)$ and $S$, estimate individual contribution
    using ICMPS
23: for $\forall v_k \in S$ do
24:     for $\forall v_x \in V_m^n(N) \setminus S$ do
25:         Calculate $p_m^n(v_k)$ using Equation 6
26:         $v_k.p := v_k.p + p_m^n(v_k)$
27:     end for
28: end for
```

Algorithm 2 shows the pheromone allocation process initiated by ant a_m in tour T_m^n. The inputs of the algorithm include tour T_m^n and number of layers of sub-network N. While the output is a set of updated pheromone. Lines 1-3 initialize the environment and a temporal variable. Lines 4-10 identify the candidates for pheromone allocation by considering the threshold ω. The N-layer sub-network is constructed through Lines 11-17. While Lines 18-22 estimate the influence contribution of each node in the tour. In the end, Lines 23-28 aim to allocate the pheromone.

The complexity of Algorithm 2 is $O(n)$. More specifically, the complexity of Lines 9-15 is $O(n)$ since N is a constant. As the seed set size is limited, the complexity of Lines 21-26 is $O(n)$.

4.4 Mining Influential Users

Mining influential users is the last step of SIMiner, which happens when the convergence of the solution emerges. It aims to identify a limited set of influential users based on the environment modified by ants, having the pheromone with different intensity distributed. Nodes with high pheromone intensity demonstrate their important position and superior influential capability in the social network. Therefore, this group of users should be first targeted and selected as seeds.

The influential users mining algorithm applied in this context is a degree-based approach, which identifies the influential users by only considering the amount of pheromone on each node. Thus, the computational complexity is merely $O(n)$.

5 Theoretical Analysis

Given a random process, i.e., $X_n = \{P(n), E(n), T(n)\}, n \in N$, the status space S is defined over a finite set of discrete decision variables $X_i, i \in \mathbb{N}$. n represents the a particular time step; $P(n)$ refers to a collection of the pheromone distribution across the entire network, where $P(n) = \{v_j.p(n) | v_j \in V\}$. $E(n)$ indicates the edge set, and $T(n)$ denotes a set of tours generated by all the ants in the $Env(n)$, where $T(n) = \{T_m^n | a_m \in A\}$.

At time step n of a dynamic network, ant a_m's transfer probability $Q(n)$ merely depends on $P(n-1)$ and $E(n)$. Whereas, tour $T(n)$ relies on $T(n-1)$ and $Q(n)$. While $P(n)$ can be determined by Equation 8. Therefore, X_{n+1} is only affected by X_n, which explicitly shows a typical Markov process.

Let $B = \{(v_i, v_j) | v_i, v_j \in T(n), v_i \in \Gamma(v_j)\}$, and v_i is followed by v_j in $T(n)$. Then, $\forall a, b \in S$, the transfer probability $Q(n)$ satisfies:

$$Q(n) = Q_n(X_{n+1} = b | X_n = a) = \prod_{(v_i, v_j) \in B} q_{ij}(n) \tag{9}$$

Thus, $Q(n)$ depends on n only. The Markov process is a non-homogeneous Markov chain.

Theorem 1. *Given $N \in \mathbb{N}^+$, $\forall n \geq N$, if $\exists q_{min}(n) > 0$, having $v_j.p(n) \geq q_{min}(n) > 0$ and $\sum_{i=1}^{\infty} \rho_i = \infty$, then the non-homogeneous Markov process $X(n) = \{P(n), E(n), T(n)\}$ converges to the optimal solution with a probability of 1.0.*

Proof. $\forall e_{ij} \in v_i.E$, $\exists e_{min}, e_{max}$, having $0 < e_{min} \leq e_{ij} \leq e_{max} \leq 1$. Then:

$$\begin{aligned} q_{ij} &= \frac{(e_{ij}.w)^\alpha \cdot (v_j.p)^\beta}{\sum\limits_{v_x \in \Gamma(v_i) \backslash T} (e_{ix}.w)^\alpha \cdot (v_x.p)^\beta} \\ &\geq \frac{(e_{min})^\alpha \cdot (v_j.p)^\beta}{\sum\limits_{v_x \in \Gamma(v_i) \backslash T} (e_{max})^\alpha \cdot (v_x.p)^\beta} \\ &\geq \frac{(e_{min})^\alpha \cdot (v_j.p)^\beta}{\sum\limits_{v_x \in \Gamma(v_i) \backslash T} (v_x.p)^\beta} = q_{min} \end{aligned}$$

According to Equation 9, $q(n) \geq (q_{min})^{|B|} > 0$. Therefore, the lower bounds of probability that Markov chain X_n converges to the optimal solution is:

$$q^*(n) = 1 - (1 - \widehat{q})^n \geq 1 - (1 - (q_{min})^{|B|})^n$$

As $0 < (q_{min})^{|B|} < 1$, therefore $0 < 1 - (q_{min})^{|B|} < 1$. When $n \to \infty$, $(1 - (q_{min})^{|B|})^n = 0$, thus, $\lim_{n \to \infty} q^*(n) = 1$ □

Lemma 2. *Given $\rho_n = 1 - \sqrt{\frac{n}{n+a}}, a \in \mathbb{N}^+$, $\sum_{n=1}^{\infty} \rho_n = \infty$*

Proof.

$$\begin{aligned} \sum_{n=1}^{\infty} \rho_i &= \sum_{n=1}^{\infty} (1 - \sqrt{\frac{n}{n+a}}) \\ &= \sum_{n=1}^{\infty} \frac{\sqrt{n+a} - \sqrt{n}}{\sqrt{n+a}} \\ &= \sum_{n=1}^{\infty} \frac{a}{n + a + \sqrt{n(n+a)}} \\ &\geq a \cdot \sum_{n=1}^{\infty} \frac{1}{2(n+a)} \\ &= \frac{a}{2} \sum_{n=1}^{\infty} \frac{1}{n+a} = \infty \end{aligned}$$

□

Theorem 3. *Let ρ_n as evaporation rate, non-homogeneous Markov chain $X(n) = \{P(n), E(n), T(n)\}$ converges to the optimal solution with a probability of 1.0.*

Proof. Suppose that v_i is not passed by any ant, the pheromone intensity of this node reaches the minimum at time step n.

$$\begin{aligned} v_j.p(n) &= \prod_{i=1}^{n} (1 - \rho_i) \cdot v_j.p(0) \\ &= \prod_{i=1}^{n} \sqrt{\frac{i}{i+a}} \cdot v_j.p(0) \\ &= \prod_{i=1}^{a} \sqrt{\frac{i}{n+i}} \cdot v_j.p(0) \end{aligned}$$

Thus, $q_{min} = \prod_{i=1}^{a} \sqrt{\frac{i}{n+i}} \cdot v_j.p(0)$

□

In Lemma 2, we proof $\sum_{n=1}^{\infty} \rho_n = \infty$, so the Markov chain satisfies the conditions in Theorem 1.

Therefore, different from the greedy algorithm with the bound of $1 - 1/e$, by utilising SIMiner, it converges to the optimal solution with a probability of 1.0 theoretically. Whereas, the total time steps of convergence is uncertain, which is affected by the choice of parameters.

6 Experiments and Analysis

Four experiments have been conducted to evaluate SIMiner. In Experiment 1, we explore the convergence of SIMiner. Experiment 2 intends to analyse the time to converge of SIMiner when giving the different size of ant agents. Experiment 3 evaluates the performance of SIMiner in static networks, and Experiment 4 analyses the behaviour of SIMiner in dynamic networks.

6.1 Experiment Setup

Datasets. The empirical analysis and evaluation have been conducted by using four public datasets as follows.

- **Ego-Facebook**[1] dataset, collected by McAuley et al. using a Facebook application [36], which is archived in Stanford Large Network Dataset Collection [37]. It contains profile and network data from 10 ego-networks, consisting of 193 circles, 4,039 users and 88,234 edges.
- **Email-Enron**[2] dataset, which covers all the email communication. It has been posted to the web by the Federal Energy Regulatory Commission [38], [39]. The Enron email network has 36,692 nodes and 367,662 Edges. To diminish the computing time, we capture a sub-graph with 10k nodes for the experiment.
- **Wiki-Vote**[3] dataset, which incorporates administrator elections and votes history data from 3 January 2008. There are 2,794 elections with 103,663 total votes and 7,066 users participating in the elections. Nodes refer to Wikipedia users and edges represent votes from one user to another [40], [41].
- **Flixster**[4] is a movie social network which enables users to publish and share the reviews on movies by rating and commenting movies. The raw dataset has been collected by Jamali et al. [42], which contains over 1 million users, 8,196,077 ratings and 7,058,819 undirected friendship affiliation links. The collected ratings range from December 2005 to November 2009. To process the raw dataset, duplicated links have been removed since the raw dataset has been crawled as a directed network. In the meanwhile, as for those users whose behaviours appear abnormal, their data have been removed. More specifically, the rating counts of such users are over 1,000 in a single day, and all the given rating scores are either 1 or 5; thus, they are most likely robots. On the other side, users with less than 100 ratings are also eliminated from the experiments.

Evaluation Metrics. To quantitatively evaluate the proposed model, we leverage two major performance metrics, i.e., the activation coverage and running time. Meanwhile, seed set variation rate is regarded as an indicator of convergence, and the corresponding indices are elaborated in Experiment 1.

- **Influence Activation Coverage** refers to the number of nodes that have been activated by the selected influential users, which represents the effectiveness of the algorithms. Specifically, by selecting the same amount of users from the same network using different selection algorithms, the higher the activation coverage, the better the performance. The IC model has been selected as the influence propagation model for evaluating the activation coverage of the selected seeds.
- **Time to Converge** describes the total time steps required by ant agents to carry out an optimal solution, at which point the convergence emerges. This means the ants already finished exploring the influencers from the environment, and it is ready for seeding process. Since SIMiner is a distributed approach, traditional estimation of running time may not be suitable.

Influence Probability and Estimation. Degree Weighted Activation (DWA) has been applied for influence probability estimation. DWA assigns the probability of each edge based on the degrees of both nodes. For example, the weight of edge is estimated as: $e_{ij}.w = 1/|\Gamma(v_j)|$.

The influence coverage of a seed set carried out by any algorithm is estimated by running numerous times of Monte-Carlo simulations.

Baselines. We compare SIMiner approach against the following algorithms in terms of influence activation coverage.

- **Greedy:** Attempt to reach the maximum influence marginal gain in selecting each seed, coming with a $(1 - 1/e)$ approximation guarantee. Greedy selection is not scalable since it relies on a large number of Monte-Carlo simulations. However, in regards to the effectiveness, i.e., global influence activation coverage, it outperforms almost all the existing improved algorithms, such as IRIE [43].
- **Degree-based Selection:** Select the users with high degree, which is based on the intuition that users with larger friend circle can influence more users in the social network.
- **Degree Discount Heuristic:** If one node is nominated as a seed, all its adjacent neighbours' node degrees are discounted by one due to the presence of the node in the seed set [2].
- **Random Selection:** Select seeds randomly. It normally performs the worst since it is not based on any heuristics.

Experiment Parameters. The default parameters for running the experiments are listed in Table 2.

6.2 Experiment 1 : Convergence Analysis

Experiment 1 aims to explore the convergence of SIMiner, showing the evolution of the solution, which is continuously optimized. In this experiment, we compute the seed set variation rate of the selected seeds over time and estimate the activation coverage of the seed set at each time step.

In SIMiner, a group of ant agents conducts global searching activities for exploring the influencers across the entire

1. http://snap.stanford.edu/data/egonets-Facebook.html
2. https://snap.stanford.edu/data/email-Enron.html
3. https://snap.stanford.edu/data/wiki-Vote.html
4. http://www.flixster.com/

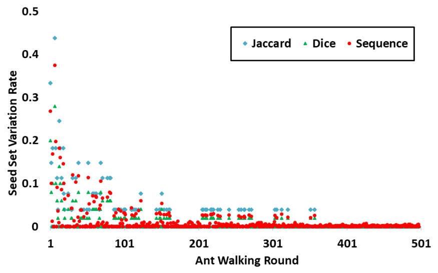

Fig. 7. Seed Set Variations (Ego-Facebook)

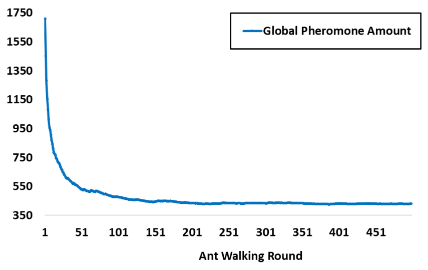

Fig. 8. Pheromone Distribution (Ego-Facebook)

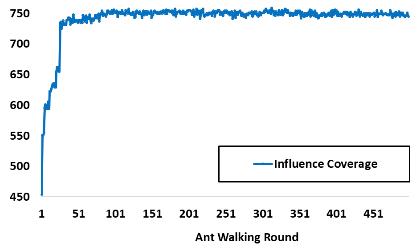

Fig. 9. Activation Coverage (Ego-Facebook)

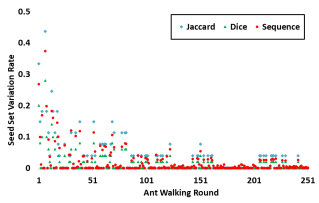

Fig. 10. One Ant (Ego-Facebook)

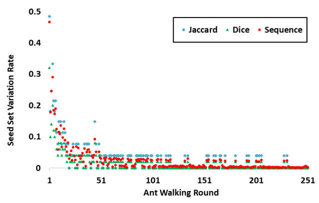

Fig. 11. Five Ants (Ego-Facebook)

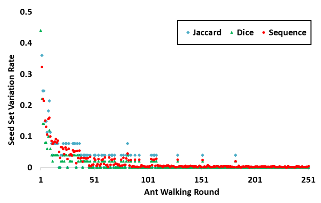

Fig. 12. Ten Ants (Ego-Facebook)

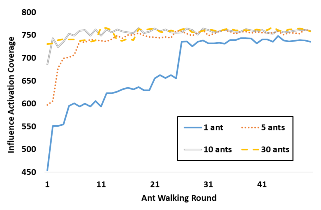

Fig. 13. Different Ant Size (Ego-Facebook)

TABLE 2
Experiment Parameters

Parameter	Default Value
Seed set size	50
Ant agents size	10
Number of time steps	500
Sub-network layer N	2
α in Equation 2	0.8
β in Equation 2	0.4
Initial pheromone amount of v_i ($v_i.p(0)$)	0.5
Threshold of skip ω	0.4
Constant D in Equation 7	1000

network. We regard the convergence emerges when the top k, i.e., the seed set size, users stay stable, without any significant seed-set variations in a number of consecutive time steps. In other words, low variation rate is a sign of convergence, and activation coverage of the seed set remains stable accordingly.

To quantify the variations between any two seed sets, three indices are adopted, i.e., Jaccard distance $d_{jcd}(S_1, S_2)$, Dice dissimilarity $d_{dic}(S_1, S_2)$ and sequential distance considering the index of the elements $d_{sqc}(S_1, S_2)$, which are formulated in Equations 10, 11 and 12, respectively. In these three equations, S_1 and S_2 denote two different seed sets, having the same cardinality, i.e., $S_1 \neq S_2, |S_1| = |S_2|$. $I(c|S_1)$ refers to the index of element c in set S_1.

$$d_{jcd}(S_1, S_2) = 1 - \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{10}$$

$$d_{dic}(S_1, S_2) = 1 - \frac{2|S_1 \cap S_2|}{|S_1| + |S_2|} \tag{11}$$

$$d_{sqc}(S_1, S_2) = \frac{1}{|S_1|}\left(\sum_{c \in S_1 \cap S_2} \frac{|I(c|S_1) - I(c|S_2)|}{|S_1|} + |S_1 \backslash S_2|\right) \tag{12}$$

In this experiment, we explore the convergence phenomenon of SIMiner by using Ego-Facebook dataset only since the same trend can be observed by using other datasets. Figure 7 demonstrates the distribution of three indices within 500 walking rounds. The seed set variation rate declines slowly with the evolutionary effect from SIMiner. Meanwhile, Figure 8 shows the global pheromone distribution trend over time. As we can observe from Figure 9 that the influence activation coverage climbs up to around 750 rapidly within 50 walking rounds, and oscillates slightly. The experimental results explicitly reveal that the solutions carried out by SIMiner evolves and eventually start to converge from a certain point.

6.3 Experiment 2 : Time to Converge

To further explore SIMiner, Experiment 2 analyses the efficiency of SIMiner based on the previous experiment. As SIMiner is a distributed approach, the efficiency evaluation and improvement are very different from other centralized algorithms. Specifically, the traditional algorithm only can be improved by modifying the algorithm itself or leverage other statistical methods. By contrast, SIMiner models the individual agent, and the overall efficiency can be improved easily by adding more ant agents. In this experiment, we analyse the efficiency of SIMiner by exploring the time to converge.

By using the Ego-Facebook dataset, we increase the number of the ant agents to explore the trend of the convergence. The results of seed set variation rate when involving one ant, five ants and ten ants are demonstrated in Figures 10, 11 and 12, respectively. It is clear that in general the more ant agents are deployed, the faster convergence can be observed.

Figure 13 compares the activation coverage trend when involving the ants of the different size in SIMiner. It is evident that a larger number of ants have a higher starting point. The time to converge is longer when only one ant is working in the environment. 30 ants and 10 ants perform almost of the same in terms of efficiency.

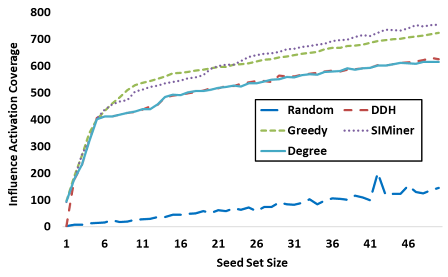

Fig. 14. Ego-Facebook

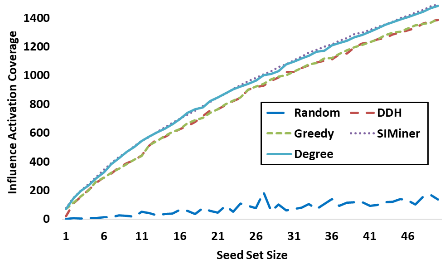

Fig. 15. Wiki Vote

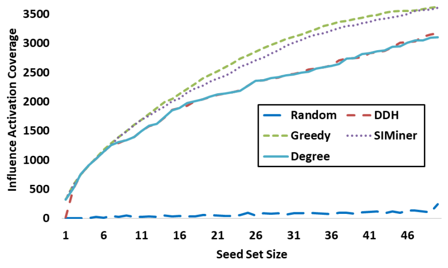

Fig. 16. Email Enron

6.4 Experiment 3 : SIMiner in Static Networks

Experiment 3 aims to evaluate the performance of SIMiner in static networks. We compare SIMiner against some other state-of-the-art algorithms using three public datasets, i.e., Ego-Facebook, Email-Enron and Wiki-Vote. Greedy selection is one of the strongest baselines in the influence maximization problem, and it outperforms most of the existing algorithms but appears not scalable.

As we can see from Figures 14, 15 and 16 that SIMiner performs similarly or even better than Greedy algorithm.

Next, the parameters of SIMiner, including ω, α and β, are adjusted to investigate the performance. Figures 17 and 18 demonstrate the results of parameter choices using Ego-Facebook dataset. No significant changes but slight differences can be observed.

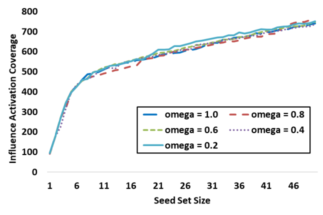

Fig. 17. Parameter Choices ω

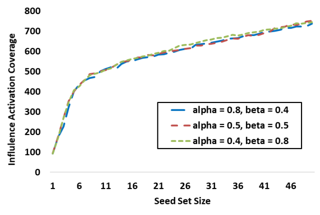

Fig. 18. Parameter Choices α, β

6.5 Experiment 4 : SIMiner in Dynamic Networks

In Experiment 4, we analysed dynamic social networks and conducted influential nodes mining from a temporal perspective. This experiment describes SIMiner handles the dynamics by reusing the pheromone accumulated at previous time step in the environment.

The experiment encompasses two scenarios, i.e., the social network keeps expanding on a monthly basis, and the size of the network varies quarterly. In both scenarios, SIMiner keeps running over time, and no interruptions occur when changing the snapshots. However, the greedy selection algorithm identifies a set of influential nodes when given the first snapshot, and is executed again after a specific time interval, so that the influential user set is recalibrated. To simulate the dynamics and evolution of social networks, we have extracted a number of snapshots from Flixster dataset by following the assumptions below.

- A user is regarded as joining the social network when giving the first rating to any movie, while he or she quits the network after contributing the last rating.

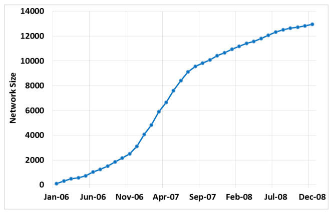

Fig. 19. Monthly Flixster Network (Incremental)

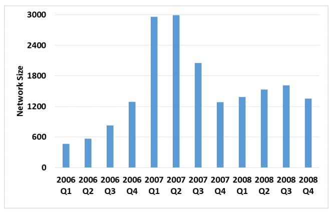

Fig. 20. Quarterly Flixster Network (Dynamic)

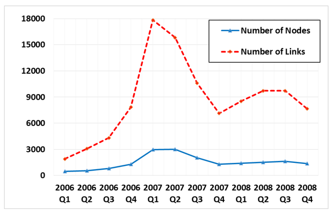

Fig. 21. Quarterly Flixster Links Variation

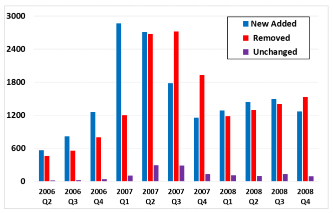

Fig. 22. Quarterly Flixster Nodes Variation

- A user is considered as effective in a particular time frame if he or she has contributed any ratings during this period, i.e., a particular snapshot.
- When users join the social network, the affiliation relationships are formed immediately.
- The influential nodes selected from a snapshot at k remain effective in the next snapshot at $k+1$.

Figures 19 and 20 demonstrate the aforementioned two scenarios, i.e., 36-month and 12-quarter snapshots respectively, ranging from the year 2006 to 2008. Figures 21 and 22 reflect the dynamic features of social networks, where the users join and quit, and links are forming and vanishing over time. We can observe from Figure 22 that the social network is actually 'changing the blood' over time, and the effective users' life cycle appears short. In Q1 2007, the difference between new joining users and quitting users is very high, while more users quit than that of joining in Q3 and Q4 2007.

The experimental results of the first scenario is presented in Figure 23, where the performance of both greedy selection and SIMiner have been compared in an incremental social network. SIMiner keeps running and selecting influential nodes. Whereas, the seed set identified by greedy algorithm is recalibrated every first month of the year. The seeds from greedy algorithm perform well for a few months. However, obvious performance degradation can be observed when the

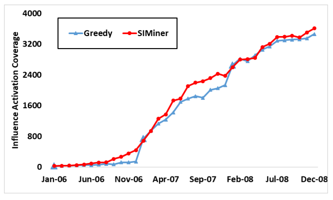

Fig. 23. SIMiner in an Incremental Environment (Monthly)

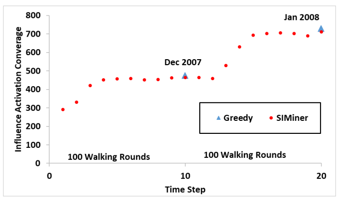

Fig. 24. SIMiner Handling Incremental Environment

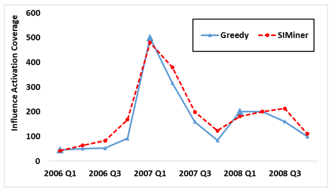

Fig. 25. SIMiner in a Dynamic Environment (Quarterly)

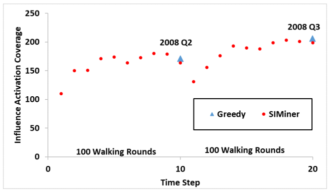

Fig. 26. SIMiner Handling Dynamic Environment

social network evolves. To examine the detailed operations of SIMiner, we select two consecutive months, i.e., December 2007 and January 2008. In Figure 24, it can be seen that when social network evolves, SIMiner is capable of adapting the solutions rapidly and carrying out an optimal solution.

Figure 25 demonstrates the results of the second scenario, where the effective users in the dynamic network vary on a quarterly basis. Greedy selection recalibrates the solution on the first quarter of each year. It is obvious that SIMiner is adaptive and performs well, whereas, without any calibration, the greedy selection loses its advantages when the social network evolves. In Figure 26, the performance of SIMiner degrades suddenly when the network snapshot changes, but increases immediately, approaching an optimal solution.

6.6 Discussion

We conducted four experiments to explore the key features of SIMiner, as well as its performance in the influence maximization problem in both static and dynamic networks. We demonstrated SIMiner's advantages in mining influencers and handling dynamics from a microscopic level. The following insights can be uncovered from the experiments.

- In the influence maximization problem, SIMiner can always converge to an optimal solution within a number of iterations. Equipped with the same knowledge, the individual's behaviours lead to a global convergence. The identified seed set is continuously being revised and optimized, and eventually reaches a stable status. SIMiner is a global searching algorithm with evolutionary computing features, which are powered by stigmergic interactions. The influence effectiveness of the seed set from SIMiner is closer or even superior to that of the Greedy algorithm. Furthermore, SIMiner demonstrates a fast convergence, implying an excellent adaptability.
- SIMiner can handle large-scale networks when big data emerge. As SIMiner is a distributed approach, inheriting from ABM, the modelling merely focuses on individual's features and behaviours. To improve the efficiency in handling large-scale networks, more agents can be simply added to expedite the convergence speed. Therefore, SIMiner is not affected by the data size.
- SIMiner requires relatively complicated parameter choices, and the selection of these parameters are dataset-dependent. In SIMiner, parameters are supposed to be carefully determined to achieve a good performance. In this paper, the selection still relies on the attempts of different combinations but we would like to set the investigation of parameter choices for SIMiner as one of the future work.
- The ant size appears to be a critical factor affecting the performance. Given a finite walking rounds within a limited time frame, employing one ant to explore a large network causes performance degradation. Because the solution may not fully converge before the walking finished. In this case, SIMiner more likely carries out a premature solution.
- SIMiner handles dynamics using agents in a local level. The environment changes are captured from a microscopic perspective. Given an undiscovered network, SIMiner requires a few time steps to "warm up", allocating pheromone to the environment, which can be regarded as an initialization. Any topological structure update is not supposed to affect the initial pheromone intensity. In other words, the exploration can be continued based on the existing pheromone and without any interruption.

7 Conclusion and Future Work

This paper presented a novel decentralized approach, i.e., SIMiner, to mine influential nodes in social networks. The outstanding merit of the proposed model is its adaptation ability. SIMiner is capable of adapting the solutions in complex dynamic environments. Specifically, a set of artificial ant agents have been deployed, which keep crawling in the network and updating the environment via stigmergic interactions, thus, the solutions keep revising with the network evolution over time. Four experiments have been conducted to analyse and evaluate the performance of SIMiner in both static and dynamic environments. Experimental results reveal that SIMiner can fast converge to an optimal solution, whose effectiveness is closer or even better than Greedy selection. Moreover, it is able to handle the influential users mining tasks in dynamic networks without any re-calibrations. Furthermore, the proposed decentralized approach is suitable for many real-world networks, as it is applicable to large-scale networks and even functions without a global view.

In the future, we plan to improve the effectiveness by granting the adaptation capabilities to each ant agent, so that they can explore the influential nodes more effectively. In addition, we may exploit a hybrid model by leveraging both centralized and decentralized approaches, where both local views from ant agents and limited global information are available. Last but not the least, we will investigate the parameter choices for SIMiner.

REFERENCES

[1] Y. Dong, Z. Ding, F. Chiclana, and E. Herrera-Viedma, "Dynamics of public opinions in an online and offline social network," *IEEE Transactions on Big Data*, vol. PP, no. 99, pp. 1–1, 2017.

[2] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Paris, France: ACM, 2009, pp. 199–208.

[3] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. San Francisco, California, USA: ACM, 2001, pp. 57–66.

[4] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. Washington, DC, USA: ACM, 2003, pp. 137–146.

[5] M. Brautbar and M. J. Kearns, "Local algorithms for finding interesting individuals in large networks," in *Innovations in Theoretical Computer Science*, 2010, pp. 188–199.

[6] C. Borgs, M. Brautbar, J. Chayes, S. Khanna, and B. Lucier, "The power of local information in social networks," in *Proceedings of the 8th international conference on Internet and Network Economics*. Springer-Verlag, 2012, pp. 406–419.

[7] C. H. Lien and Y. Cao, "Examining wechat users motivations, trust, attitudes, and positive word-of-mouth: Evidence from china," *Computers in Human Behavior*, vol. 41, pp. 104–111, 2014.

[8] D. Shiffman, S. Fry, and Z. Marsh, *The nature of code*. D. Shiffman, 2012, ch. 7 Cellular Automata, pp. 323–330.

[9] M. Dorigo, E. Bonabeau, and G. Theraulaz, "Ant algorithms and stigmergy," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 851–871, 2000.

[10] E. Bonabeau, "Editor's introduction: stigmergy," *Artificial Life*, vol. 5, no. 2, pp. 95–96, 1999.

[11] A. Mostafa, M. Zhang, and Q. Bai, "Trustworthy stigmergic service composition and adaptation in decentralized environments," *Service Computing, IEEE*, vol. 9, no. 2, pp. 317 – 329, January 2014.

[12] W. Li, Q. Bai, C. Jiang, and M. Zhang, "Stigmergy-based influence maximization," in *Proceedings of Pacific Rim International Conference on Artificial Intelligence (PRICAI)*. Phuket, Thailand: Springer, 2016, pp. 750–762.

[13] D. Ye, M. Zhang, and A. V. Vasilakos, "A survey of self-organization mechanisms in multiagent systems," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, vol. PP, no. 99, pp. 1 – 21, January 2016.

[14] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th International Conference companion on World Wide Web*. Hyderabad, India: ACM, 2011, pp. 47–48.

[15] H. Zhang, D. T. Nguyen, H. Zhang, and M. T. Thai, "Least cost influence maximization across multiple social networks," *IEEE Transactions on Network Science and Engineering*, vol. PP, no. 99, pp. 1–11, 2015.

[16] P. Holme, "Modern temporal network theory: a colloquium," *The European Physical Journal B*, vol. 88, no. 9, pp. 1–30, 2015.

[17] B. Wang, G. Chen, L. Fu, L. Song, X. Wang, and X. Liu, "DRIMUX: dynamic rumor influence minimization with user experience in social networks," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 2016, pp. 791–797.

[18] G. Song, Y. Li, X. Chen, X. He, and J. Tang, "Influential node tracking on dynamic social network: an interchange greedy approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 2, pp. 359–372, 2017.

[19] Y. Bao, X. Wang, Z. Wang, C. Wu, and F. Lau, "Online influence maximization in non-stationary social networks," in *Proceedings of IEEE/ACM International Symposium on Quality of Service*, Beijing, China, 2016.

[20] G. Tong, W. Wu, S. Tang, and D.-Z. Du, "Adaptive influence maximization in dynamic social networks," *IEEE/ACM Transactions on Networking*, 2016.

[21] N. T. Gayraud, E. Pitoura, and P. Tsaparas, "Diffusion maximization in evolving social networks," in *Proceedings of the 2015 ACM on Conference on Online Social Networks*. ACM, 2015, pp. 125–135.

[22] F. Karimi and P. Holme, "Threshold model of cascades in empirical temporal networks," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 16, pp. 3476–3483, 2013.

[23] M. Gomez-Rodriguez, L. Song, N. Du, H. Zha, and B. Schölkopf, "Influence estimation and maximization in continuous-time diffusion networks," *ACM Transactions on Information Systems (TOIS)*, vol. 34, no. 2, p. 9, 2016.

[24] N. Ohsaka, T. Akiba, Y. Yoshida, and K.-i. Kawarabayashi, "Dynamic influence analysis in evolving networks," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1077–1088, 2016.

[25] K. Wang, L. Huang, C. Zhou, W. Pang *et al.*, "Traveling salesman problem," in *Proceedings of Conference on Machine Learning and Cybernetics*, vol. 3, Xi'an, China, 2003, pp. 1583–1585.

[26] J. Takahashi, R. Kanamori, and T. Ito, "A preliminary study on anticipatory stigmergy for traffic management," in *Proceedings of the 2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology(IAT 2012)*, vol. 3. Macau, China: IEEE, 2012, pp. 399–405.

[27] Hadeli, P. Valckenaers, M. Kollingbaum, and H. Van Brussel, "Multi-agent coordination and control using stigmergy," *Computers in industry*, vol. 53, no. 1, pp. 75–96, 2004.

[28] T. G. Lewis, "Cognitive stigmergy: A study of emergence in small-group social networks," *Cognitive Systems Research*, vol. 21, pp. 7–21, 2013.

[29] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Link mining: models, algorithms, and applications*. Springer, 2010, pp. 337–357.

[30] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 2, 2007.

[31] G. Theraulaz and E. Bonabeau, "A brief history of stigmergy," *Artificial life*, vol. 5, no. 2, pp. 97–116, 1999.

[32] Y. Jiang and J. Jiang, "Diffusion in social networks: A multiagent perspective," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 45, no. 2, pp. 198–213, 2015.

[33] Y. Wang, G. Cong, G. Song, and K. Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Washington, DC, USA.: ACM, 2010, pp. 1039–1048.

[34] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4, pp. 175–308, 2006.

[35] P. Kumar and G. Raghavendra, "A note on the parameter of evaporation in the ant colony optimization algorithm," in *International Mathematical Forum*, vol. 6, no. 34, 2011, pp. 1655–1659.

[36] J. J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks." in *NIPS*, vol. 2012, 2012, pp. 548–56.

[37] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun 2014.

[38] B. Klimt and Y. Yang, "Introducing the enron corpus." in *CEAS*, 2004.

[39] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.

[40] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 641–650.

[41] ——, "Signed networks in social media," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1361–1370.

[42] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM Conference on Recommender Systems*. ACM, 2010, pp. 135–142.

[43] K. Jung, W. Heo, and W. Chen, "Irie: Scalable and robust influence maximization in social networks," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 918–923.