An Efficient Memetic Algorithm for Influence Maximization in Social Networks

Maoguo Gong, Chao Song, Chao Duan, Lijia Ma, and Bo Shen
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, CHINA

Abstract–Influence maximization is to extract a small set of nodes from a social network which influences the propagation maximally under a cascade model. In this paper, we propose a memetic algorithm for community-based influence maximization in social networks. The proposed memetic algorithm optimizes the 2-hop influence spread to find the most influential nodes. Problem-specific population initialization and similarity-based local search are designed to accelerate the convergence of the algorithm. Experiments on three real-world datasets demonstrate that our algorithm has competitive performances to the comparing algorithms in terms of effectiveness and efficiency. For example, on a real-world network of 15233 nodes and 58891 edges, the influence spread of the proposed algorithm is 12.5%, 13.2% and 173.5% higher than the three comparing algorithms Degree, PageRank and Random, respectively.

I. Introduction

In recent years, online social network sites such as Facebook and Twitter have enjoyed an increasing attention. They are becoming popular platforms selected by companies for promoting their products and spreading information with the number of users growing rapidly. In these websites, users are enthusiastic about interacting, sharing and collaborating through online social media which makes information spread among users more easily [1]–[4]. The word-of-mouth marketing or viral marketing, an effective strategy on social network marketing, aims to produce a maximal cascading effect in social networks by targeting only a small number of selected users. Specifically, to promote a new product in a social network, the pre-existing adopters will make a recommendation to their friends, and then their friends will make a recommendation to their friends' friends and so on. This strategy is proved to be cost-effective and successful [5]–[6]. However, the crucial problem is how to select

Digital Object Identifier 10.1109/MCI.2016.2572538
Date of publication: 18 July 2016

Corresponding Author: Maoguo Gong
(E-mail: gong@ieee.org).

1556-603X/16©2016IEEE

Influence maximization is to extract a small set of nodes (or seeds) from a social network which can generate the propagation maximally under a cascade model.

IMAGE LICENSED BY INGRAM PUBLISHING

the pre-existing adopters who may produce the largest influence in a social network.

The crucial problem above, known as influence maximization, is to extract a small set of nodes (or seeds) from a social network which can generate the propagation maximally under a cascade model [1], [7]–[14]. This problem is first studied by Domingos and Richardson [6]. They view a market as a social network and propose a probabilistic method with modeling the influence between network users as a Markov random field. Then the problem is formalized as a discrete optimization problem which is proved to be NP-hard under the independent cascade model (IC) and the linear threshold model (LT) by Kempe et al. [15]. A natural hill-climbing greedy strategy is also proposed by them as an approximate algorithm to solve the optimization problem. However, the greedy algorithm has two intrinsic difficulties to be applied to large social networks [7]. Firstly, the greedy algorithm needs to traverse all remaining nodes in the network when it selects each next seed [7]. As a result, the computation of the greedy algorithm has a quadratic relationship with the number of nodes [7]. Secondly, computing exact influence spread of a node set under the IC model and the LT model is #P-hard [7]. Therefore, a Monte-Carlo simulation is run to obtain an accurate estimation. However, this needs a large number of runs, typically 10,000 times, which results in a large computation time.

Memetic Algorithms (MAs) [16], a branch of evolutionary computation, are hybrids of global search methods and local search procedures [17]–[21]. The global search methods are generally evolutionary and swarm intelligence [17]. They can produce a reliable estimate of the global optimum. The local search procedures are individual refinement processes incorporating domain-knowledge. They can explore better solutions around the best solution found so far [22]–[23]. MAs have been proved to play an important role in solving complex optimization problems in social networks. In [24], a memetic algorithm named as Meme-Net is proposed to solve the resolution limit problem in community detection. In [25], a fast memetic algorithm is presented to solve community detection effectively. The algorithm adopts a multi-level learning strategy as the local search procedure and shows its effectiveness. In [26], a fast memetic algorithm is proposed to compute and transform structural balance in signed networks.

In this paper, we propose a novel memetic algorithm for influence maximization in social networks, termed as CMA-IM. Fig. 1 provides the framework of CMA-IM, which comprises three steps: (1) Network clustering; (2) Candidate selection; (3) Seed generation. In the first step, we use a fast two-phase heuristic algorithm BGLL [27] to detect communities in networks. In the second step, we first select the communities that are significant and then choose a few nodes from each significant community to form the candidate pool. In the final step, we model the influence maximization problem as the optimization of a 2-hop influence spread [13] and propose a problem-specific memetic algorithm to find the ultimate seeds. The proposed algorithm combines a genetic algorithm as the global search method and a similarity-based strategy as the local search procedure. Experiments on several real-world networks show that CMA-IM can get competitive results in terms of effectiveness and efficiency.

The contributions of our work are as follows:

- We propose a novel memetic algorithm for the community-based influence maximization problem. To the best of our knowledge, the proposed algorithm is the first attempt for dealing with the influence maximization by a memetic algorithm. Memetic algorithms have already been proved to be effective in dealing with complex optimization problems, so solutions generated by the proposed memetic algorithm may be close to the optimal solution.

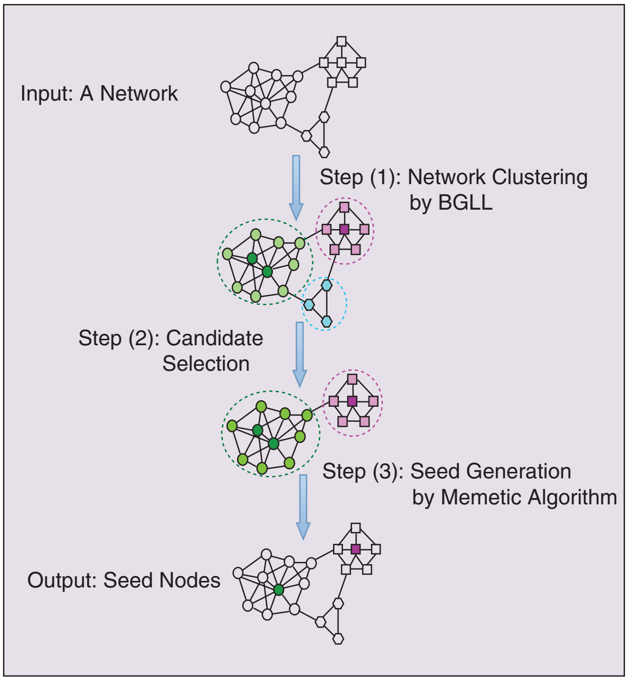
Input: A Network
Step (1): Network Clustering by BGLL
Step (2): Candidate Selection
Step (3): Seed Generation by Memetic Algorithm
Output: Seed Nodes

FIGURE 1 The framework of CMA-IM. In Step (1), the input network is divided into communities by network clustering algorithm BGLL. In Step (2), candidates are selected from each significant community. And in Step (3), the ultimate seed set is determined by the proposed memetic algorithm.

- We propose a problem-specific population initialization method and a similarity-based local search procedure, which can accelerate the convergence of the algorithm. In order to solve the problem of influence overlapping, we also apply the similarity-based strategy to the processes of crossover and mutation.
- We estimate the effectiveness and the efficiency of CMA-IM on real-world networks. The experimental results show that our algorithm has competitive performances to the comparing algorithms.

The rest of this paper is organized as follows: Section II reviews related works. Section III gives a description of the problem model. Section IV presents our proposed CMA-IM algorithm in detail. Section V shows experimental performances of the proposed algorithm on three real-world networks. Finally, the conclusion is summarized in Section VI.

II. Related Works

Many methods have been proposed for influence maximization [1], [7], [9]–[12]. Leskovec et al. [9] present a method called CELF which is reported to be 700 times faster than the greedy algorithm. They develop a lazy procedure when selecting new seeds. The lazy procedure exploits the submodularity of the spread function to reduce the number of function evaluations. Goyal et al. [7] propose an improvement on [9] named as CELF++ which further improves the efficiency. Chen et al. [10] put forward two new greedy algorithms. One is the NewGreedy. It attempts to reduce computations by generating a new smaller graph with all edges not participating in the propagation removed. The other one is the MixedGreedy. It runs the NewGreedy in the first round and runs the CELF [9] in the later rounds. Although these improvements outperform the greedy algorithm in efficiency, they still cannot be scalable to large-scale networks. Recently, some authors use the communities of the networks to improve the efficiency of algorithms. Wang et al. [11] introduce a community-based greedy algorithm called CGA which narrows down the search space of the influential nodes from the whole graph to the communities. They exploit dynamic programming to select the community which has the largest increase of influence spread and adopt MixGreedy algorithm [10] to find the most influential node as the seed in the chosen community. Chen et al. [1] propose a community-based algorithm under the Heat Diffusion Model, termed as CIM. They select seeds from the candidate nodes by comparing scores of them. Rahimkhani et al. [12] present a fast algorithm, called ComPath, combining with the community character and introduce an influence spread estimation under the LT model.

In this paper, network clustering is completed by BGLL which is different from the network clustering algorithms used in [1] and [12]. In [1], the authors propose a hierarchical clustering (HClustering) method which iteratively merges nodes into communities based on the structural similarity between each pair of nodes. In [12], the authors employ the SLPA algorithm [28] which discovers overlapping communities according to the listener-speaker interaction rules.

III. Problem Model

We model a social network as an undirected network denoted as $G=(V,E)$ where V represents the node set denoting users in the social network and E represents the edge set denoting the relationships between users [10]. N and M are the number of nodes and edges, respectively.

The proposed memetic algorithm optimizes the 2-hop influence spread to find the most influential nodes. Problem-specific population initialization and similarity-based local search are designed.

Given a node set S that includes k nodes, the influence spread produced by S which is denoted as $\sigma(S)$ is the number of nodes that S can influence in the network G under a certain cascade model. In this paper, we select the IC model as the cascade model, which is widely used in the previous works [7], [9], [10], [13], [15]. In the IC model, the state of a node has only two types, either active or inactive. The inactive nodes can be changed into the active nodes, but not vice versa. Each edge is associated with a propagation probability p and $p(u,v)$ represents the probability of an inactive node v to be influenced by its active neighbor u. Under the IC model, $\sigma(S)$ works as follows. Let S_t be the set of nodes that are active in the step t, and $S_0 = S$ which is initialized by a k-node set. At step t, each node $v \in S_{t-1}$ has only one chance to independently activate every inactive neighbors with the probability p. This influence diffusion process stops at the step t when $S_t = \varnothing$ and $\sigma(S)$ is the union of S_t obtained at each step. Thus, the influence maximization problem is to find the k-node set S that can make $\sigma(S)$ maximal under the IC model. Let us illustrate this problem using the greedy algorithm as an example on a toy network in Fig. 2.

Suppose that we attempt to find a 2-node set S that can influence the most number of nodes of the network in Fig. 2. When it is to find the first seed, the greedy algorithm traverses all the nodes from 1 to 10 and calculates the influence spread of each node that is estimated under the IC model. It selects node 6 which has the maximum influence spread as the first seed. To find the second seed, the greedy algorithm computes the influence spread of (6, 1), (6, 2),..., (6, 5), (6, 7),..., (6, 10) and eventually it selects node 3 as the second seed, because it results in the most increase of the influence spread. The process to find the 2-node set $S=\{6,3\}$ which possesses the maximal influence is the influence maximization problem. Here, the influence spread of a node set is calculated by running 10,000 times a Monte-Carlo simulation.

As mentioned above, computing exact $\sigma(S)$ under the IC model is #P-hard and it needs to run a number of Monte-Carlo simulations. And with the size of a social network growing larger, the time consumed by running Monte-Carlo simulation becomes not negligible.

In [13], Lee et al. propose a fast approximation for influence maximization, they consider the influence spread on the nodes within 2-hops away from seed node set instead of all nodes in the network. The 2-hop influence spread of a node set $\hat{\sigma}(S)$ is calculated as (1).

$$\hat{\sigma}(S)=\sum_{s\in S}\hat{\sigma}\{s\}-\Big(\sum_{s\in S}\sum_{c\in C_s\cap S} p(s,c)(\sigma_c^1-p(c,s))\Big)-\chi, \quad (1)$$

where C_s denotes the 1-hop nodes cover of node s, i.e., the neighbors of node s, p is the propagation probability in the IC model, $\chi=\sum_{s\in S}\sum_{c\in C_s\cap S}\sum_{d\in C_c\cap S\setminus\{s\}} p(s,c)p(c,d)$ and $\sigma_c^1 = 1+\sum_{c\in C_u} p(u,c)$. The term σ_c^1 is the 1-hop influence spread of node c. In (1), the first term is the sum of the 2-hop influence spread of each seed in S, the second term considers the redundant situation that a seed is a neighbor of another seed and the third term considers the redundant situation that a seed is 2-hops away from another seed.

The authors show that the 2-hop influence spread is sufficiently valid and efficient to estimate the influence spread of a node set. Therefore, we adopt the 2-hop influence spread in our algorithm. The important variables mentioned above are listed in Table 1.

IV. The Proposed Algorithm for Influence Maximization

In this section, we will introduce our proposed algorithm CMA-IM. As illustrated earlier in Fig. 1, CMA-IM consists of three steps: (1) Network clustering; (2) Candidate selection; (3) Seed generation. A detailed description of each step will be given in the following.

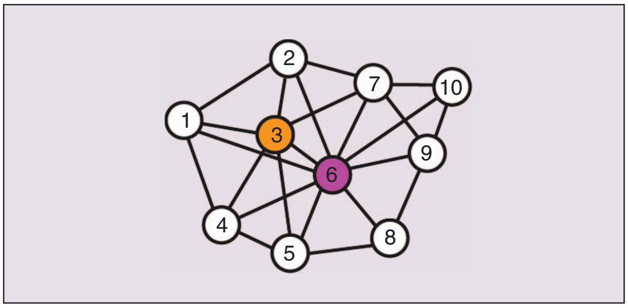

FIGURE 2 An illustration for the influence maximization problem using the greedy algorithm on a toy network. Node 6 is the first selected seed and node 3 is the second selected node.

TABLE 1 The variables used in this paper.

VARIABLES	DESCRIPTIONS
$G(V, E)$	An undirected network with node set V and edge set E
N	The number of nodes in G
M	The number of edges in G
k	The number of seeds $(1 \le k \le N)$
S	The seed set with k-node
p	Propagation probability

Social networks naturally tend to be clustered into groups or communities. The nodes connect more densely with the nodes in the same group than the nodes outside the group.

A. Network Clustering

Social networks naturally tend to be clustered into groups or communities [29]. The nodes connect more densely with the nodes in the same group than the nodes outside the group. Clustering into communities is a property of social networks, which is beneficial for understanding the structure of networks [30]. Modularity, a famous criterion proposed by Newman and Girvan [29], provides a quality evaluation of the network community structure. Various algorithms for network clustering are based on modularity optimization including heuristic algorithms [27], [31]–[33] and evolutionary algorithms [24], [30].

BGLL proposed by Blondel et al. [27] is a fast heuristic method based on modularity optimization, which consists of two phases. At the first phase, each node of the network is considered as a community. Then they remove a node from its original community to its neighbor's community which has the maximal positive gain in modularity. This phase is applied repeatedly for all nodes until no further improvement can be achieved. The first phase is then complete [27]. The second phase considers the communities obtained in the first phase as nodes such that a new network can be built. Then BGLL runs these two phases iteratively until achieving an unchanged result and obtaining the maximal modularity.

Compared with the HClusreing and SLPA algorithm, the BGLL algorithm can discover more natural structures of networks because it needs no prior knowledge about the community number. So communities obtained by the BGLL get closer to the inherent communities in networks. Meanwhile BGLL only needs a few iterations to obtain a maximal modularity, which makes the BGLL algorithm have better performance in terms of efficiency when applied to large-scale networks.

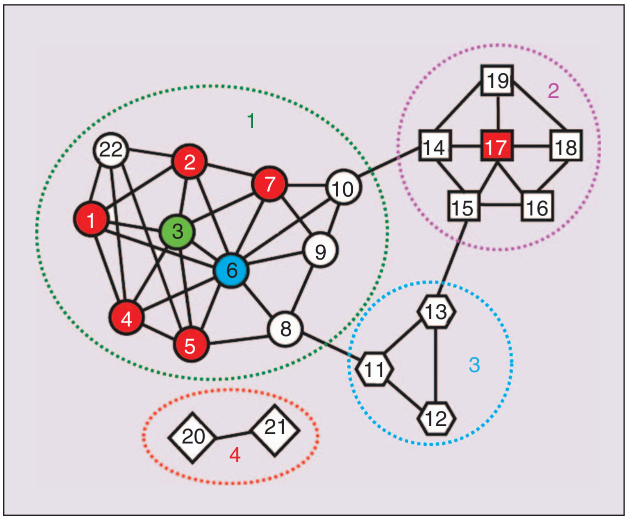

FIGURE 3 A network with three communities. The communities are numbered as 1, 2 and 3, respectively. The red nodes are common neighbors of the node 3 and the node 6.

B. Candidate Selection

The candidate selection step aims to determine a set of candidate nodes according to the information about communities obtained in the first step. Because social networks in realistic settings are usually extremely huge, the search space for selecting seeds is also huge. Therefore, there is a need to effectively reduce the number of candidate nodes.

By analyzing the structures of communities, we find that not all communities are significant enough to accommodate seed nodes. For example, in Fig. 3, although the network is divided into three communities, community 3 may be insignificant compared with community 1 and 2 due to its smaller community size. Suppose we choose a seed node from community 3, it may only activate three nodes initially. We choose the community 1 and 2 as the significant communities because their sizes are large and there may be more influential nodes in them. Here, we define significant communities as the first n large communities, where n is varied with networks. However, the nodes in community 1 have many common neighbors. For example, node 3 and node 6 have 5 common neighbor nodes, which is not beneficial to influence spread. In order to solve the problem of overlapping influence, we propose a similarity-based high degree method called SHD which is described in Section IV-C1.

Let *Candidate* be a candidate node pool. Next, our task is to choose a number of potential nodes from each significant community to fill in the *Candidate*. If considering the influence ability of a node, degree centrality may be the most intuitive way to estimate the ability. The higher degree of a node is, the more neighbors the node has, which means that node with higher degree can influence more nodes with the same propagation probability. Therefore, we select potential nodes from each significant community based on the degree centrality. Here, we choose the way of [12], which is shown as (2), to decide the number of candidate nodes selected from each significant community.

$$\left(\frac{C_i - MinC}{MaxC - MinC}\right) * \beta + \alpha, \qquad (2)$$

where C_i is the size of the i-th significant community, $MaxC$ is the size of the largest significant community and $MinC$ is the size of the smallest significant community. The term $(C_i - MinC)/(MaxC - MinC)$ is the ratio of the i-th community among all selected communities and its value is confined to $[0,1]$. β is the amplification term and α is the constant term that guarantees the least selection in each community. After selecting the *Candidate*, the last step of our algorithm is to generate the ultimate seeds.

C. Seed Generation

After the two aforementioned steps, the search space has been reduced. In the next step, we will employ the proposed problem-specific memetic algorithm, named as Meme-IM, to generate the ultimate seeds by optimizing the 2-hop influence spread shown as (1). The whole framework of Meme-IM is shown as **Algorithm 1**.

In **Step 1)**, Meme-IM mainly completes the population initialization task. Firstly, it creates the initial population of solutions $P = \{x_1, x_2, \ldots, x_{\text{pop}}\}^T$ according to a problem-specific strategy. And then it selects the individual with the maximum fitness as P_{best}. **Step 3)** is the evolution procedure. In **Step 3.1)**, Meme-IM first uses the deterministic tournament selection method to select parental individuals P_{parent} for mating in genetic algorithm. Then in **Step 3.2)**, Meme-IM reproduces the chosen parental individuals P_{parent}, i.e., performs crossover and mutation operation on P_{parent}. **Step 3.3)** is an individual reinforcement procedure. **Step 3.4)** is to refresh the current population by taking the best *pop* individuals from $P \cup P_{\text{new}}$. And in **Step 4)**, when the algorithm terminates on convergence, Meme-IM stops and outputs the ultimate *k*-node set.

Algorithm 1 **Framework of Meme-IM.**

```
Input: Maximum generation: maxgen, population size: pop, mating pool size: pool, tournament size: tour, crossover probability: pc, mutation probability: pm, spread probability: p, seed size: k, the candidate nodes pool: Candidate and the connection matrix: A.
Output: The most influential k-node set.
Step 1) Initialization
Step 1.1) Population initialization:
    $P = \{x_1, x_2, \ldots, x_{\text{pop}}\}^T$;
Step 1.2) Best individual initialization: $P_{\text{best}} = x_i$;
Step 2) Set $t = 0$; // the number of generations
Step 3) Repeat
Step 3.1) Select parental chromosomes for mating;
    $P_{\text{parent}} \leftarrow$ Selection(P, pool, tour);
Step 3.2) Perform genetic operators:
    $P_{\text{child}} \leftarrow$ GeneticOperation($P_{\text{parent}}$, pc, pm);
Step 3.3) Perform local search:
    $P_{\text{new}} \leftarrow$ LocalSearch($P_{\text{child}}$);
Step 3.4) Update population:
    $P \leftarrow$ UpdatePopulation(P, $P_{\text{new}}$);
Step 3.5) Update the best individual $P_{\text{best}}$;
Step 4) Stopping criterion: If $t < maxgen$, then
    $t = t + 1$ and go to Step 3), otherwise, stop the algorithm and output.
```

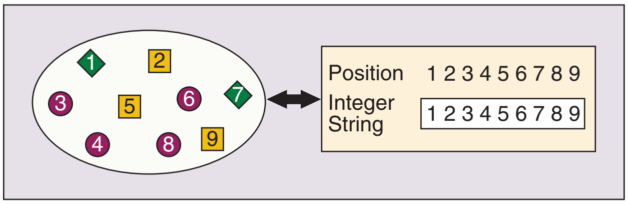

FIGURE 4 Illustration of the representation. Left: a 9-node set selected from *Candidate*. Right: the individual encoding of the 9-node set.

In the following, we will give a more detail description of several important procedures including the population initialization, genetic operation and local search procedure.

1) Representation and Initialization

In Meme-IM, each chromosome (individual) x_a $(1 \le a \le pop)$ in the population represents a *k*-node set, which is encoded as an integer string

$$x_a = \{x_a^1, x_a^2, \ldots, x_a^k\},$$

where *k* is the number of seeds, each gene x_a^i of the chromosome corresponds to a node selected from *Candidate*. An illustration of this representation is shown as Fig. 4. It is noticed that there is no repeated node in x_a. Considering that the solution produced by selecting *k* nodes randomly from *Candidate* is of low quality and may result in a long time to converge, we attempt to initialize a higher quality population to speed up the convergence. Here, we propose a similarity-based high degree method called SHD. The SHD and the random mechanism can guarantee the convergence and diversity of the individuals. The population initialization procedure is shown as **Algorithm 2**.

High degree centrality is a standard method for influence maximization on social and other networks [34]. But high

Algorithm 2 **Population initialization.**

```
Input: Population size: pop
Output: Population P
Generate a half of population based on SHD, see
    Algorithm 3 for more information;
for i from 1 to (pop/2) do
    for j from 1 to k do
        if $rand(1) > 0.5$ then
            select a random node different from each node in $x_i$
            from the Candidate to replace $x_i^j$;
        end if
    end for
end for
for i from $(pop/2 + 1)$ to pop do
    select k different nodes from the Candidate to
    initialize $x_i$ based on SHD;
end for
```

degree centrality may produce overlapping influence spread between nodes. To solve this problem, we propose a similarity-based high degree method (SHD). SHD starts with choosing the node with the highest degree in *Candidate*. After choosing a node, it excludes the neighbor nodes that are similar with the existing nodes. Then SHD chooses the next node with the highest degree in the left candidate nodes. This procedure iteratively operates until k nodes are chosen. Here, the degree of similarity between nodes is measured by the structure similarity defined as (3). The process is described as **Algorithm 3**.

In **Algorithm 3**, $N(v)=\{\exists u \in V, uv \in E\}$ represents the neighbors of node v and the structural similarity between nodes u and v is defined as (3).

$$Similarity(u,v)=\frac{|NB(u)\cap NB(v)|}{|NB(u)|+|NB(v)|}, \tag{3}$$

where $NB(v)=\{v \mid v \cup N(v)\}$ includes the node v and its neighbors. In **Algorithm 3**, *sim* is a threshold confined to [0, 1] which is set according to different datasets. When the structure similarity between two nodes is larger than *sim*, the nodes are similar. The similarity between nodes is a significant criterion to present from overlapping influence spread.

2) Genetic Operators

Crossover

In Meme-IM, we employ the one-point crossover because it is simple. The one-point crossover works as follows. Given two parent chromosomes x_a and x_b, we first randomly select a crossing over position i $(1 \le i \le k)$, then exchange each node j after the position i between the two parents $(i.e., x_a^j \leftrightarrow x_b^j, \forall j \in \{j \mid i \le j \le k\})$, and then two new offspring chromosomes x_c and x_d return. We also should guarantee the validity of x_c and x_d, i.e., there are no same nodes in x_c and x_d, respectively. Specifically, when the j-th node in x_b (x_a) is not similar with the nodes in x_a (x_b) except for x_a^j (x_b^j), then we exchange x_b^j and x_a^j.

Algorithm 3 SHD algorithm.

```
Start with $x_a = \varnothing$;
$TempCandidate = Candidate$;
for $i$ from 1 to $k$ do
    choose a node $v_i \in TempCandidate$ with the highest degree;
    $x_a \leftarrow x_a \cup \{v_i\}$;
    $SimNeighbor \leftarrow \{u \in N(v) \mid Similarity(u,v) \ge sim\}$;
    $TempCandidate \leftarrow$
    $\{v \mid v \in TempCandidate, v \notin SimNeighbor, v \ne v_i\}$;
    if $TempCandidate = \varnothing$ do
        $x_a \leftarrow x_a \cup \{v_{i+1}, v_{i+2}, \ldots, v_k\}$, $v_{i+1}, v_{i+2}, \ldots, v_k$ are selected from $Candidate$ randomly;
        break;
    end if
end for
return $x_a$
```

Mutation

Here, we employ the similarity-based mutation on the generated population after crossover. For each gene in a chromosome, if the generated random value $r \in [0,1]$ is smaller than the mutation probability pm, then we mutate the gene to another gene in *Candidate*. The other gene is selected randomly from the genes which are dissimilar with the genes in the chromosome. The similarity is evaluated by (3). However, when the value of r is larger than pm, there is also a situation we mutate the gene. When the gene is similar with the most influential gene in the chromosome, we mutate the gene to another gene in *Candidate* which is dissimilar with the most influential gene and its similar neighbors.

3) Local Search Procedure

Local search procedure is an individual reinforcement procedure which is to find a better solution around the best solutions found so far [23]. For a chromosome, when we change a node to another node in the *Candidate* which is different from any node in the chromosome, a neighbor of the chromosome is obtained. Here, we employ a similarity-based strategy as the local search procedure. The procedure is performed on an arbitrary individual in the population, then it attempts to find a better individual from the neighborhoods of the individual. The fitness is calculated by (1) and the neighbor individual is better if its fitness is larger than that of the original individual. If the change can produce a better individual, this change is accepted. The procedure repeats until no further improvement can be made. Here, we find the fittest chromosome in P_{child} which is obtained after the genetic operators and apply the local search procedure on it. The implementation of the local search procedure is shown as **Algorithm 4**.

In **Algorithm 4**, we first perform FindBest() function to select the individual with the maximum fitness in the input individuals. Then we apply the local search procedure on it.

Algorithm 4 The local search procedure.

```
Input: $P_{child}$
Output: $P_{child}$
$N_{current} \leftarrow FindBest(P_{child})$;
$islocal \leftarrow FALSE$;
repeat
    $N_{next} \leftarrow FindBestNeighbor(N_{current})$;
    if $Eval(N_{next}) > Eval(N_{current})$;
        $N_{current} \leftarrow N_{next}$;
    else
        $islocal \leftarrow TRUE$;
    end if
until $islocal$ is $TRUE$
```

The Eval() function is to compute the fitness of a solution based on (1). The FindBestNeighbor() function is to find the best neighbor chromosome with the largest fitness value.

The similarity-based local search procedure attempts to find a better individual from the neighborhoods of the individual.

V. Experimental Study

In this section, we evaluate the effectiveness and the efficiency of our proposed algorithm CMA-IM and compare the influence spread and the running time with other six algorithms on three real-world social networks.

A. Experiment Setting

1) Datasets

Dolphin network [35]. The small size Dolphin social network describes the associations between 62 bottlenose dolphins living in Doubtful Sound, New Zealand [35]. Lusseau observed the behavior of the dolphins in a period of seven years. The nodes of the network represent dolphins and edges represent a statistically frequent associations between these dolphins.

NetGRQC network [36]. The medium size NetGRQC network is a collaboration network whose nodes represent authors and edges represent co-authors relationships between them. If two authors coauthor a paper, an edge establishes between them. The data contains papers collected from the "General Relativity and Quantum Cosmology" section of the e-print arXiv (http://www.arXiv.org) in a period from January 1993 to April 2003.

NetHEPT network [10]. The large size NetHEPT network is also a collaboration network of paper co-authors and is frequently used in previous works such as [10]. The papers in this dataset are obtained from the "High Energy Physics-Theory" section of the e-print arXiv from year 1991 to year 2003. This network contains multiedges if the two co-authors have collaborated multiple papers.

The basic characteristics of the real-world networks described above are given in Table 2.

2) Comparing Algorithms

CGA-IM and MA-IM. CGA-IM is the variant version of CMA-IM by removing the local search procedure and MA-IM is the variant of CMA-IM by removing the network clustering step. We compare CMA-IM with CGA-IM and MA-IM on two real-world networks to demonstrate the effectiveness of the network clustering and the local search procedure in CMA-IM.

CELF. The CELF algorithm [9] is an improvement on the greedy algorithm which has the same result as the greedy algorithm and as much as 700 times speedup. Here, we take the number of Monte-Carlo simulations as 10,000 to obtain an accurate estimate.

CMA-HClustering and CMA-SLPA. CMA-HClustering and CMA-SLPA are two comparing algorithms combining our memetic algorithm and the network clustering algorithms in [1] and [12], respectively.

Degree centrality. The Degree centrality [15] is a classical heuristic for mining influential nodes, which believes the more neighbor nodes connected, the more important the node is. It sorts nodes in the descending order according to their degrees and simply selects the top k nodes as the seeds.

PageRank. PageRank is the popular algorithm for ranking webpages based on their importance proposed by Brin and Page [37]. Here, the restart parameter is set as 0.15 and the stop criterion is set as 0.001. PageRank sorts the nodes according to their importance and return their ranks. We select the top k nodes as seed nodes.

Random. Random is a baseline method used in [15]. The random method randomly selects k nodes as the seeds.

Other famous heuristics are not taken into consideration such as the distance centrality and the betweeness centrality because of their high computational cost [10].

All experiments are implemented under the IC model. In order to compare the accuracy of different algorithms, we compute the influence spread of the ultimate k-node set of each algorithm by running Monte-Carlo simulation for 10,000 times and take the average influence spread. All algorithms are independently run 30 times on each network. All the experiments are conducted on a PC with 1.70 GHz Inter Core i5 and 8.00 GB Memory. The experimental parameters of our algorithm are listed in Table 3.

TABLE 2 Statistics of the three real-world networks.

NETWORK	NODES	EDGES	AVERAGE DEGREE
DOLPHIN	62	159	5.129
NETGRQC	5242	14496	5.5261
NETHEPT	15233	58891	3.8635

TABLE 3 The parameters in our algorithm.

PARAMETER	MEANING	VALUE
α	The constant term in (2)	4
β	The amplification term in (2)	10
maxgen	The maximum generation	50
pop	Population size	200
pool	Size of the mating pool	100
tour	Tournament size	2
pc	Crossover probability	0.8
pm	Mutation probability	0.2

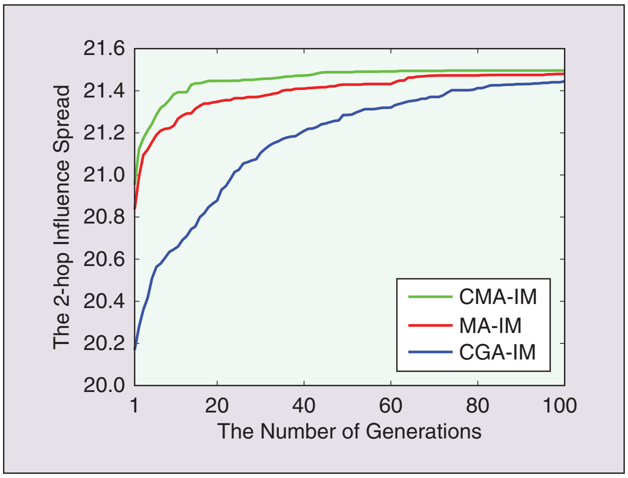

FIGURE 5 Comparisons between CMA-IM and its variants CGA-IM and MA-IM in terms of convergence on the Dolphin social network, respectively.

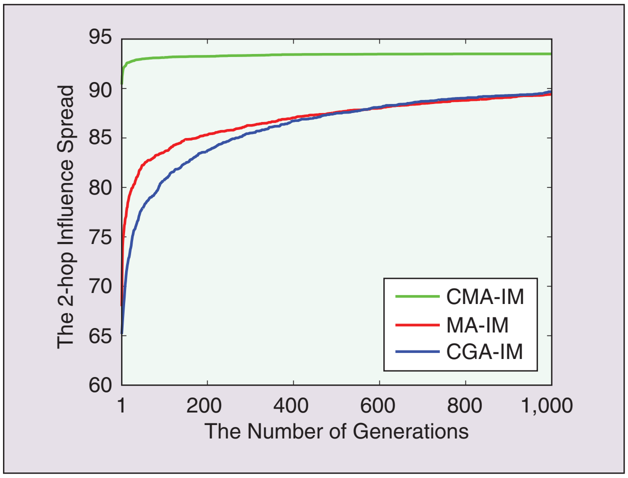

FIGURE 6 Comparisons between CMA-IM and its variants CGA-IM and MA-IM in terms of convergence on the NetHEPT network, respectively.

TABLE 4 The information about communities and candidates of the three networks.

NETWORK	DOLPHIN	NetGRQC	NetHEPT
Nodes	62	5242	15233
Seeds	10	30	30
Communities	5	392	1820
Significant communities	5	35	50
Candidates	47	227	315

B. Experiments on Real-World Networks

In order to compare the convergence of CMA-IM with its two variants CGA-IM and MA-IM, we set the parameter *maxgen* larger than the original setting. However, the parameters in each pair of comparisons remain the same.

Firstly, the comparison between CMA-IM and its variant CGA-IM is made to show the effectiveness of the local search procedure. We test CMA-IM and CGA-IM on two real-world networks, the Dolphin social network and the NetHEPT network. In these experiments, we set *maxgen* as 100 for the Dolphin network and 1,000 for the NetHEPT network. The results are shown as the green and blue lines in Figs. 5 and 6.

The green and blue lines in Fig. 5 show the 2-hop influence spread obtained by CMA-IM and CGA-IM with generation increasing from 1 to 100 on the Dolphin social network. And the green and blue lines in Fig. 6 show the 2-hop influence spread obtained by CMA-IM and CGA-IM with generation increasing from 1 to 1,000 on the NetHEPT network. For the small Dolphin social network, when the generation is up to 100, the 2-hop influence spread produced by CMA-IM and CGA-IM does not differ a lot from each other. Both of them can reach an optimal solution. CMA-IM with local search can converge within 50 generations while CGA-IM without local search needs more generations. However, for the large NetHEPT network, CGA-IM cannot find the optimal solution within 1,000 generations while CMA-IM can evolve to a better solution efficiently within 50 generations. These results demonstrate that local search can speed up the convergence and produce a higher quality solution, especially when the search space is large.

Next, the comparison between CMA-IM and its variant MA-IM is made to illustrate the effectiveness of the network clustering step. We test CMA-IM and MA-IM on two real-world networks, the Dolphin network and the NetHEPT network. The parameter *maxgen* is set as 100 for the Dolphin network and 1,000 for the NetHEPT network. The results are shown as the green and red lines in Figs. 5 and 6.

The green and red lines in Fig. 5 show the 2-hop influence spread obtained by CMA-IM and MA-IM with generation increasing from 1 to 100 on the Dolphin social network. The green and red lines in Fig. 6 show the 2-hop influence spread obtained by CMA-IM and MA-IM with generation increasing from 1 to 1,000 on the NetHEPT network. In this paper, we apply network clustering step to narrow down the search space of the seed nodes. Table 4 gives the number of candidates resulted after reducing the search space. For the Dolphin social network, we take all the five communities as significant communities. The search space is not reduced obviously, so MA-IM achieves a similar result as that of the CMA-IM. However, CMA-IM can converge within 50 generations while MA-IM needs more than 60 generations. For the NetHEPT network, the influence spread of CMA-IM is much better than that of MA-IM because the search space is reduced nearly 50 times. When the generation increases up to 1,000, MA-IM still cannot achieve the optimal solution. However, CMA-IM only needs less than 50 generations. Therefore, the combination of the network clustering and the memetic algorithm is effective that can improve the performance of our algorithm apparently.

Finally, the comparisons between CMA-IM and other six state-of-the-art algorithms are made. We compare the influence

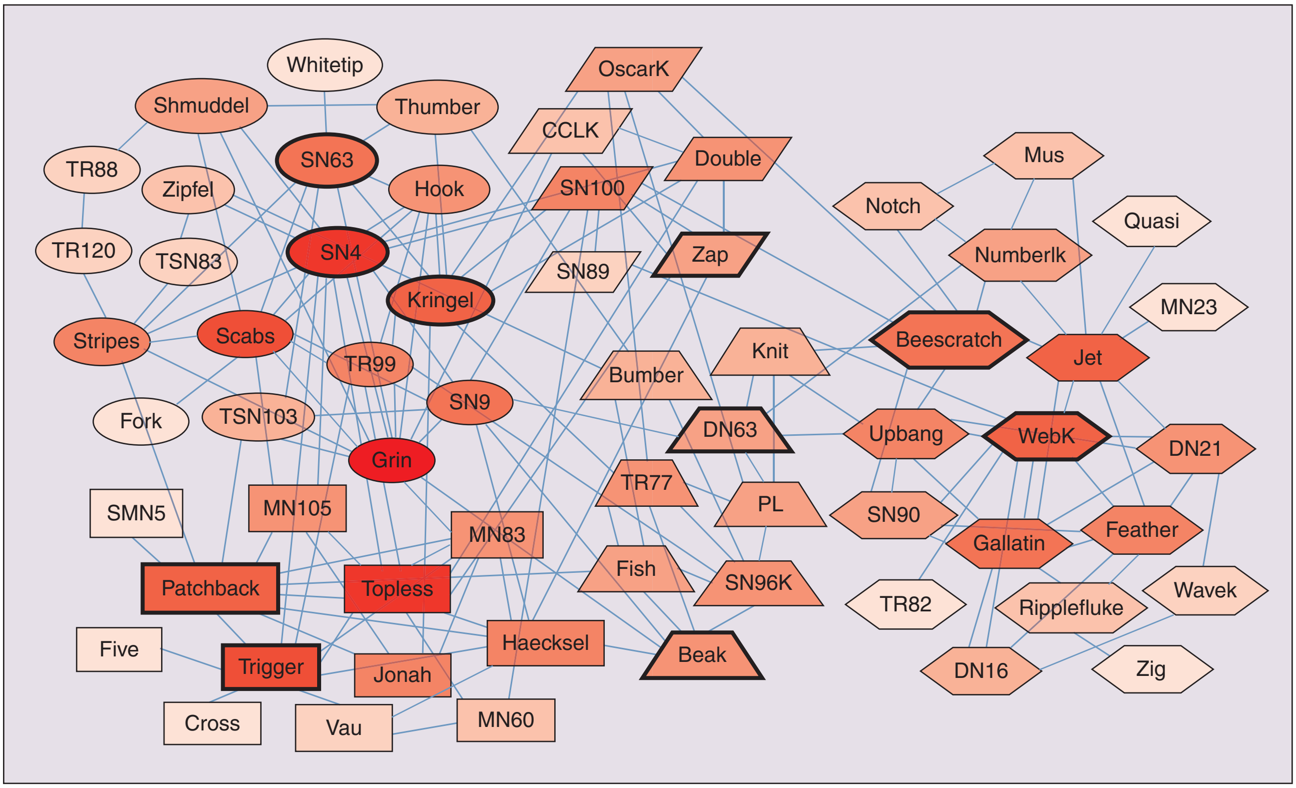

FIGURE 7 An illustration for the community structure and the ultimate seeds of the Dolphin network. The Dolphin network clusters into five communities, the nodes of the same shape belong to the same community. The color of nodes from dark to light corresponds to the degree of nodes from large to small. The ten bold nodes are the seeds generated by CMA-IM.

spread and the running time of each algorithm with the number of seeds increasing on the three real-world networks. In these experiments, the parameter *maxgen* is set as 50. Figs. 8 to 10 show the influence spread of seven algorithms on the three networks under the IC model whose x-axis represents the seed set size and y-axis represents the influence spread estimated by running Monte-Carlo simulation for 10,000 times. And Fig. 11 shows the running time of seven algorithms whose x-axis represents seven algorithms on the three networks and y-axis represents the running time of each algorithm in log scale. The percentages below about influence spread are computed when the seed set size is 30 (The seed set size is 10 for the case of the Dolphin network).

Fig. 7 is an illustration for the community structure and the ultimate seeds of the Dolphin network. From Fig. 7, we can see that the network is partitioned into five communities by BGLL algorithm and the nodes of the same shape belong to the same community. The color of nodes from dark to light red corresponds to the degree of nodes from large to small. The bold nodes are the ultimate seeds generated by CMA-IM. The network is low scale and connected sparsely, and we set the seed set size ranging from 1 to 10 and set the propagation probability p as 0.1.

Fig. 8 shows the influence spread on the Dolphin network. From Fig. 8, we can see that the differences in the influence spread of seven algorithms are not obvious. The result of CELF greedy algorithm is the best and CMA-IM essentially matches CELF. The results of CMA-IM, CMA-HClustering and CMA-SLPA are extremely close. Compared with other heuristics, CMA-IM is 2.3%, 4.6% and 21.4% better than Degree centrality, PageRank and the baseline method Random, respectively.

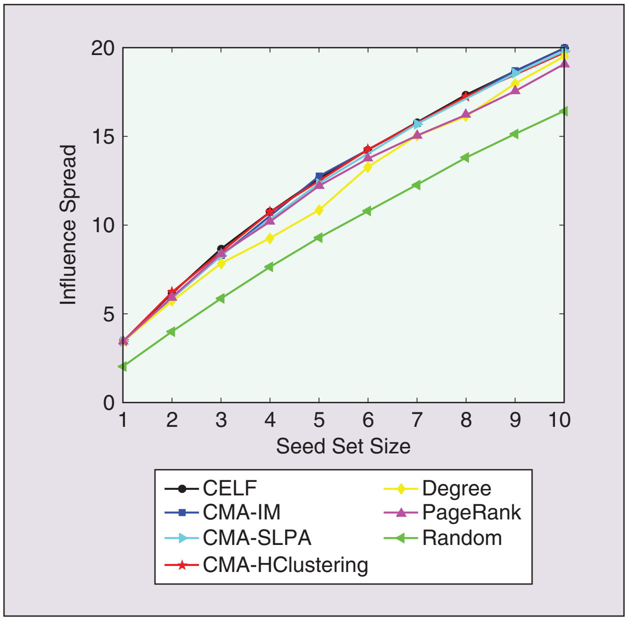

FIGURE 8 Influence spread of different algorithms on the Dolphin network ($N = 62$, $M = 159$, and $p = 0.1$).

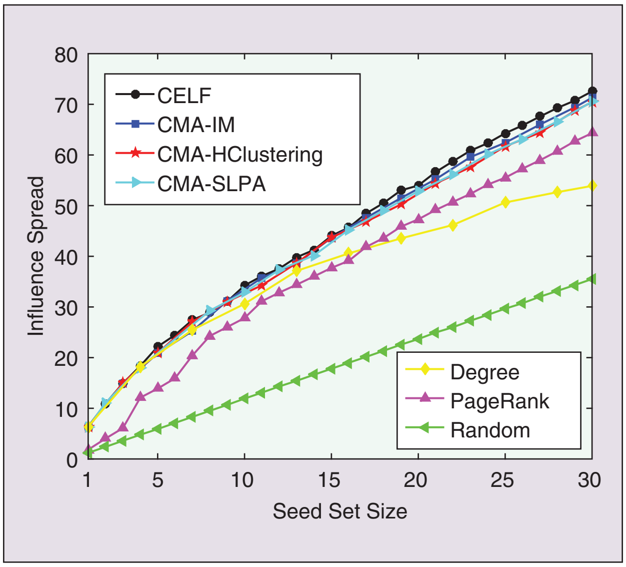

FIGURE 9 Influence spread of different algorithms on the NetGRQC network ($N = 5242$, $M = 14496$, and $p = 0.01$).

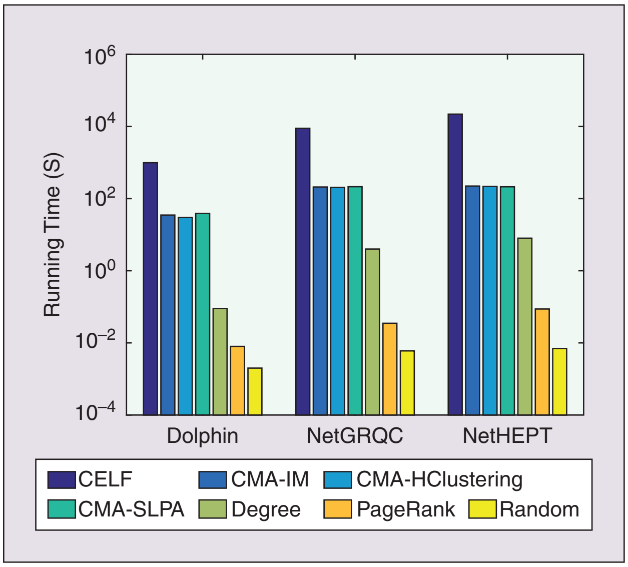

FIGURE 11 The running time of different algorithms for three real-world networks.

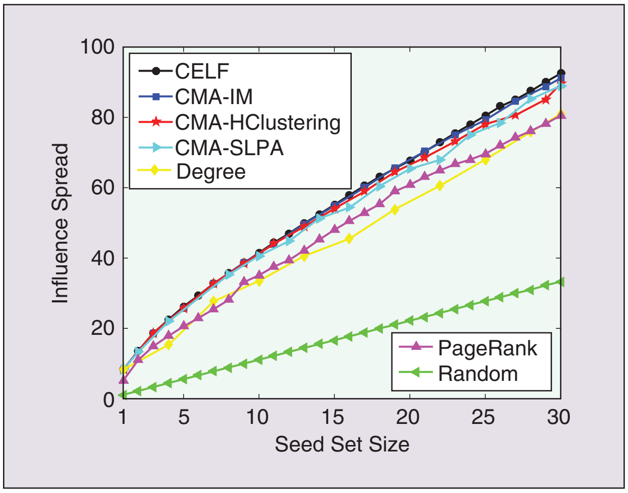

FIGURE 10 Influence spread of different algorithms on the NetHEPT network ($N = 15233$, $M = 58891$, and $p = 0.01$).

In Fig. 7, we can see that our generated seeds belong to five communities while the nodes with top 10 degree belong to three larger communities. It shows that selecting many top degree nodes in large communities may not influence more nodes. Clustering network into communities and selecting a number of top degree nodes in each significant community can get efficient candidate nodes. For the running time, Fig. 11 shows that the CELF is quite slow, CMA-IM is one order of magnitude better than CELF. CMA-IM, CMA-HClustering and CMA-SLPA have close running time. The heuristic algorithms outperform CMA-IM in running time while their performances are poor.

Fig. 9 shows the influence spread on the NetGRQC network. The result in Fig. 9 indicates that CELF produces the largest influence spread and the result of CMA-IM is close to that of CELF with 1.72% lower. CMA-IM outperforms CMA-HClustering and CMA-SLPA slightly. And CMA-IM outperforms the three heuristics, Degree, PageRank and Random with 32.3%, 10.9% and 100.8% higher, respectively. We find that the nodes with top 30 degree are only within two communities while the seeds discovered by CMA-IM cover twelve communities. As a result that Degree centrality produces overlapping influence and limits the influence spread. It can be seen from Fig. 9, the influence spread increases more and more gently with selecting more and more top degree nodes. This shows CMA-IM can reduce the overlapping influence spread and find more influential nodes. For the running time, when the network grows larger, the low effectiveness of CELF becomes apparent. It takes hours to find 30 seeds while CMA-IM only needs minutes.

Fig. 10 shows the influence spread on the NetHEPT network. From Fig. 10, we can see that the influence spread produced by CMA-IM almost matches that of CELF with 1.5% lower. The results of the three community-based algorithms are still close to each other. CMA-IM is 12.5%, 13.2% and 173.5% higher than Degree, PageRank and Random, respectively. When looking at the running time, CELF takes two orders of magnitude longer time than CMA-IM. From the comparison between the running time of CMA-IM on NetGRQC and NetHEPT, we can see that the running time on NetHEPT is close to NetGRQC due to the network clustering step which reduces the search space of candidate nodes efficiently.

As summarized from the experimental comparisons, the proposed memetic algorithm plays an important role in speeding up the convergence and finding the promising solutions in a low running time. For the running time, the Random has the best performance. The Degree centrality and PageRank also perform better than CMA-IM. However, the Degree centrality, PageRank and Random cannot provide a seed set with good quality. Although the CELF method can provide a reliable seed set for influence maximization, it is not scalable for large-scale networks. The proposed CMA-IM algorithm can solve the problem of the influence maximization both effectively and efficiently on social networks with different sizes.

VI. Conclusion

In this paper, an efficient memetic algorithm for information maximization has been proposed. The community property has been incorporated to reduce the search space of seed nodes effectively. Then a problem-specific memetic algorithm has been proposed to optimize the 2-hop influence spread which can estimate the influence spread of a node set effectively. In the memetic algorithm, we design a problem-specific population initialization method and a similarity-based local search procedure, which can accelerate the convergence of the algorithm. The similarity between nodes is taken into consideration to solve the problem of influence overlapping. The experiments on three real-world networks illustrate that the proposed CMA-IM algorithm has a good performance in terms of the effectiveness and efficiency on social networks with different sizes.

Due to the increasing scale of the social networks, there is a need to make influence maximization algorithms more efficient. As a possible future work, we will consider to investigate how to extend CMA-IM to a parallel framework. We will also develop our CMA-IM algorithm to other information cascade models such as the linear threshold model and the weighted cascade model.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant nos. 61273317, 61422209, 61473215), the National Program for Support of Top-notch Young Professionals of China, the Specialized Research Fund for the Doctoral Program of Higher Education (Grant no. 20130203110011) and the Fundamental Research Fund for the Central Universities (Grant no. K5051202053).

References

[1] Y. C. Chen, W. Y. Zhu, W. C. Peng, W. C. Lee, and S. Y. Lee, "CIM: Community-based influence maximization in social networks," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 2, pp. 25, 2014.

[2] E. Cambria, H. Wang, and B. White, "Guest editorial: Big social data analysis," *Knowledge-Based Syst.*, vol. 69, pp. 1–2, 2014.

[3] M. Grassi, E. Cambria, A. Hussain, and F. Piazza, "Sentic web: A new paradigm for managing social media affective information," *Cogn. Computat.*, vol. 3, pp. 480–489, 2011.

[4] X. Han, W. Wei, C. Miao, J. P. Mei, and H. Song, "Context-aware personal information retrieval from multiple social networks," *IEEE Computat. Intell. Mag.*, vol. 9, no. 2, pp. 18–28, 2014.

[5] E. Cambria, M. Grassi, A. Hussain, and C. Havasi, "Sentic computing for social media marketing," *Multimedia Tool. Applicat.*, vol. 59, pp. 557–577, 2012.

[6] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, 2001, pp. 57–66.

[7] A. Goyal, W. Lu, and L. V. Lakshmanan, "CELF++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proc. 20th Int. Conf. Companion on World Wide Web*, Hyderabad, India, 2011, pp. 47–48.

[8] I. Chaturvedi and J. C. Rajapakse, "Building gene networks with time-delayed regulations," *Pattern Recogn. Lett.*, vol. 31, pp. 2133–2137, 2010.

[9] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Jose, CA, 2007, pp. 420–429.

[10] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 199–208.

[11] Y. Wang, G. Cong, G. Song, and K. Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Washington, DC, 2010, pp. 1039–1048.

[12] K. Rahimkhani, A. Aleahmad, M. Rahgozar, and A. Moeini, "A fast algorithm for finding most influential people based on the linear threshold model," *Expert Syst. Applicat.*, vol. 42, no. 3, pp. 1353–1361, 2015.

[13] J.-R. Lee and C.-W. Chung, "A fast approximation for influence maximization in large social networks," in *Proc. Companion Publication of the 23rd Int. Conf. World Wide Web Companion*, Seoul, Korea, 2014, pp. 1157–1162.

[14] B. Liu, G. Cong, Y. Zeng, D. Xu, and Y. M. Chee, "Influence spreading path and its application to the time constrained social influence maximization problem and beyond," *IEEE Trans. Knowledge Data Eng.*, vol. 26, no. 8, pp. 1904–1917, 2014.

[15] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Washington, DC, 2003, pp. 137–146.

[16] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," Caltech Concurrent Computation Program, C3P Rep., vol. 826, 1989.

[17] X. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 591–607, 2011.

[18] L. Jiao, M. Gong, S. Wang, B. Hou, Z. Zheng, and Q. Wu, "Natural and remote sensing image segmentation using memetic computing," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 78–91, 2010.

[19] Z. Zhu, S. Jia, and Z. Ji, "Towards a memetic feature selection paradigm [application notes]," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 41–53, 2010.

[20] J. Wu, Z. Chang, L. Yuan, Y. Hou, and M. Gong, "A memetic algorithm for resource allocation problem based on node-weighted graphs [application notes]," *IEEE Comput. Intell. Mag.*, vol. 9, no. 2, pp. 58–69, 2014.

[21] J. Zhang, Z. Zhan, Y. Lin, N. Chen, Y. Gong, J. Zhong, H. S. H. Chung, Y. Li, and Y. Shi, "Evolutionary computation meets machine learning: A survey," *IEEE Comput. Intell. Mag.*, vol. 6, no. 4, pp. 68–75, 2011.

[22] Y. S. Ong, M. H. Lim, and X. Chen, "Research frontier-memetic computation–past, present & future," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–31, 2010.

[23] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4135–4151, 2011.

[24] M. Gong, B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks," *Phys. Rev. E*, vol. 84, no. 5, pp. 056101, 2011.

[25] L. Ma, M. Gong, J. Liu, Q. Cai, and L. Jiao, "Multi-level learning based memetic algorithm for community detection," *Appl. Soft Comput.*, vol. 19, pp. 121–133, 2014.

[26] L. Ma, M. Gong, H. Du, B. Shen, and L. Jiao, "A memetic algorithm for computing and transforming structural balance in signed networks," *Knowledge-Based Syst.*, vol. 85, pp. 196–209, 2015.

[27] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, pp. P10008, 2008.

[28] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops*, Vancouver, BC, 2011, pp. 344–349.

[29] M. E. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, 2006.

[30] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 82–97, 2014.

[31] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, no. 6, pp. 066111, 2004.

[32] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Natl. Acad. Sci.* vol. 105, no. 4, pp. 1118–1123, 2008.

[33] M. J. Barber and J. W. Clark, "Detecting network communities by propagating labels under constraints," *Phys. Rev. E*, vol. 80, no. 2, pp. 026129, 2009.

[34] R. Albert, H. Jeong, and A. L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[35] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405, 2003.

[36] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowledge Disc. Data*, vol. 1, no. 1, pp. 2, 2007.

[37] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comp. Networks ISDN Syst.*, vol. 30, no. 1, pp. 107–117, 1998.