Knowledge-Based Systems 187 (2020) 104833

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks[☆]

Jianxin Tang [a,b], Ruisheng Zhang [a,*], Ping Wang [a], Zhili Zhao [a], Li Fan [a], Xin Liu [a]

[a] School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China
[b] School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China

ARTICLE INFO

Article history:
Received 11 November 2018
Received in revised form 30 June 2019
Accepted 4 July 2019
Available online 8 July 2019

Keywords:
Social networks
Viral marketing
Influence maximization
Discrete shuffled frog-leaping algorithm
Swarm intelligence

ABSTRACT

Influence maximization problem aims to select a subset of k most influential nodes from a given network such that the spread of influence triggered by the seed set will be maximum. Greedy based algorithms are time-consuming to approximate the expected influence spread of given node set accurately and not well scalable to large-scale networks especially when the propagation probability is large. Conventional heuristics based on network topology or confined diffusion paths tend to suffer from the problem of low solution accuracy or huge memory cost. In this paper an effective discrete shuffled frog-leaping algorithm (DSFLA) is proposed to solve influence maximization problem in a more efficient way. Novel encoding mechanism and discrete evolutionary rules are conceived based on network topology structure for virtual frog population. To facilitate the global exploratory solution, a novel local exploitation mechanism combining deterministic and random walk strategies is put forward to improve the suboptimal meme of each memeplex in the frog population. The experimental results of influence spread in six real-world networks and statistical tests show that DSFLA performs effectively in selecting targeted influential seed nodes for influence maximization and is superior than several state-of-the-art alternatives.

1. Introduction

Social networks have become powerful platforms for information diffusion and viral marketing by expanding billions of loyal users. An underlying cause fostering the capabilities is the social influence, which maps the interactions between individuals in the network and can be evaluated based on trust and reputation [1]. One of the typical applications promoted by social network is the viral marketing [2], which appreciates the important effect of 'word-of-mouth' that indwells the interpersonal influence relationship of consumers and can reshape consumers' attitudes and behaviors [3]. Influence maximization problem is targeted to select a subset of k influential seed nodes that can maximize the spread of influence into the network. The problem was coined by Domingos and Richardson [4] firstly in terms of network perspective through which the most potential customers are identified to maximize the expected profit of a product promotion activity.

As emphasized in [5,6], there are two challenges in tackling influence maximization problem. The first difficulty is to estimate the influence spread of given node set accurately, which was proved to be ♯P-hard. The second one is to provide effective and efficient algorithms for the selection of a subset influential nodes which can maximize the spread of influence into the network. Kempe et al. [7] firstly formulated influence maximization as a discrete optimization problem and proposed a greedy approach with guaranteed solution accuracy. However, experimental results [8,9] showed that greedy algorithm is time-consuming especially in large-scale networks. This is because the algorithm has to run k rounds to select the targeted seed nodes. In each round, the algorithm needs to carry out R ($R \geq 10,000$) Monte-Carlo simulations to evaluate the marginal gain of each of the N nodes in the network approximately, and for each simulation the M edges of the network will be traversed inevitably. Consequently, the time complexity of the original greedy algorithm is $O(kNMR)$.

Following up on the seminal work, novel influence estimators and influential node selecting approaches have emerged to solve influence maximization problem in a more efficient way. Chen [10] proposed an improved greedy algorithm by pruning the edges that hardly take part in influence spread in the network. Jiang et al. [11] proposed an expected diffusion value estimator to evaluate the spread of influence within the one-hop area of given candidate nodes. However, it performs less effective than the local influence estimator that optimizes the expected influence spread within the two-hop area of given candidate nodes [12].

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.07.004.

* Corresponding author.
E-mail address: zhangrs@lzu.edu.cn (R. Zhang).

https://doi.org/10.1016/j.knosys.2019.07.004
0950-7051/

Kimura et al. [13] assumed that influence only spreads along the shortest and second shortest paths, and proposed a shortest path-based influence maximization algorithm. Further more, by assuming that influence spreads on the paths independent of each other, Kim et al. [14] proposed a parallel influence path-based algorithm to identify the seed nodes in a faster way. Cao et al. [15] systematically studied the influence maximization problem based on community detection. They transformed influence maximization to an optimal resource allocation problem and proposed an optimal dynamic programming algorithm to find an optimal seed allocation. As demonstrated in [16], community-based influence maximization algorithms are generally faster than traditional greedy algorithms, but the accuracy and the scalability of the community-based algorithms need improved. Compared with the original simple greedy algorithm, those methods are more efficient by reducing or avoiding the number of Monte-Carlo simulations. However, sacrifices in solution accuracy and memory cost have to be made to compensate these novel influence maximization algorithms. Therefore, developing effective and efficient methods for influence maximization in large-scale networks still remains as an open research topic of social network analysis and is of great significance due to its promising applications in the spread of information, such as innovation diffusion, viral marketing, etc.

The effectiveness and robustness of meta-heuristic algorithms based on swarm intelligence have been widely validated by many applications on complex optimization problems such as symbolic regression problem [17], feature selection in data mining and machine learning [18], sports training sessions [19] as well as influence maximization problem [20,21], etc. In this paper, a discrete shuffled frog-leaping algorithm (DSFLA) is proposed based on network topology characteristic to identify influential nodes for influence maximization. The main contributions of our paper are as follows.

- Encoding mechanism for virtual frog individual and discrete evolutionary rules for frog population are conceived based on network topology structure, respectively. Then the framework of discrete shuffled frog-leaping algorithm for influence maximization problem is presented.
- To facilitate the global exploratory solution during the evolutionary process, a local exploitation mechanism combining deterministic and random walk strategies is put forward to improve the suboptimal meme of each memeplex in the frog population.
- The orthogonal experimental design method is adopted to optimize the parameter settings of DSFLA, and the experimental results and statistical tests in six real-world networks show that the proposed DSFLA is effective and efficient for influence maximization, and can be scalable to large-scale networks.

The remainder of this paper is organized as follows: Section 2 reviews related works. Influence maximization problem, the independent cascade model and an effective influence estimator used in this paper are introduced in Section 3. Section 4 gives the proposed discrete shuffled frog-leaping algorithm and the framework of DSFLA for influence maximization. Performance validation of DSFLA and statistical tests are provided in Section 5. Section 6 concludes this paper with future works.

2. Related works

Since the seminal work by Domingos and Richardson [4], great attention has been paid to the interesting problem. In general, the existing majority of influence maximization algorithms can be mainly categorized into the following three aspects: greedy based algorithms, reverse influence sampling algorithms and advanced heuristic algorithms.

2.1. Greedy based algorithms

Kempe and Kleinberg [7] firstly formulated influence maximization as a discrete optimization problem and proved it to be NP-hard under the independent cascade (IC) model and linear threshold (LT) model. Meanwhile, a hill-climbing greedy algorithm, which can achieve a guarantee with errors bounded at $(1-1/e-\epsilon)$ to the optimal solution approximately, was proposed to select the most influential nodes. However, the simple greedy algorithm is time-consuming because tens of thousands Monte-Carlo simulations have to be conducted in each round to select the seed node with the maximal marginal gain. By further exploring the property of submodularity, Leskovec et al. [8] proposed the CELF algorithm to improve the efficiency of the simple greedy algorithm by leveraging an influence estimation priority queue and a *lazy-forward* strategy. Based on the fact that the spread of individual's influence tends to be localized in a numbered friends of local area and one's social communication is always clustered into several steady groups [22], Kundu and Pal [9] proposed a deprecation based greedy strategy (DGS) for community networks. The influence of each candidate node is approximated by an integrated centrality metric in its appurtenant community and nodes with lower influence spread are marked to be deprecated, yet DGS is still time-consuming in large-scale networks. Shang et al. [16] adopted the seed nodes expansion strategy to approximate the expected influence of a node within its own community and proposed a community-based framework CoFIM for influence maximization. To improve the efficiency of the expansion strategy in large-scale networks, Shang et al. [23] further promoted the algorithm and derived a novel influence estimator based on multi-neighbor potential of node in community networks. To relieve the huge time consumption of the natural greedy algorithm, Lu et al. [24] proposed a CascadeDiscount algorithm to estimate node's marginal gain by removing its influence loss on neighbors from its initial influence evaluated by a ScoreCumulate model, then the most influential seed nodes are selected into the seed set based on greedy strategy.

Compared with the original greedy algorithm, improved greedy algorithms can show good performance on pruned networks at small propagation probabilities. However, they suffer from the problem of highly time consumption easily when the network scale or the propagation probability is large.

2.2. Reverse influence sampling algorithms

Random sampling theory has found its applications in influence maximization problem in the last few years. Borgs et al. [25] firstly proposed a fast reverse influence sampling (RIS) method based on random sampling theory to estimate the expected influence spread of an influential node. According to the ideology of RIS, if a node appears often as an "influencer", then it is likely a good candidate for the most influential node. Theoretical analysis proved that the method can obtain a near-optimal approximation factor of $(1-1/e-\epsilon)$ in nearly optimal time.

To optimize the number of generated subgraphs and sampling scheme, modified methods were proposed to alleviate the deficiencies of the basic RIS algorithm. TIM+ [26] and SKIM [27] were proposed to select seed nodes incrementally using a concept of combined reachability sketch, but they need huge amounts of memory cost for the generated subgraphs. Two novel sampling frameworks naming SSA and D-SSA were proposed by Nguyen et al. [28] recently, which are up to 1200 times faster than the IMM method [29] while providing the same $(1-1/e-\epsilon)$ approximation guarantee according to the experimental results. Though the sampling-based algorithms can select the k influential seed nodes efficiently in large-scale networks, they suffer the problem of huge memory cost inevitably because large amounts of RIS samples have to be generated to provide an approximately guarantee especially in large-scale networks. In addition, experiments showed that the method tends to achieve suboptimal solutions with the increase of the targeted seed set size [30].

2.3. Advanced heuristic algorithms

Conventional heuristic methods, including high degree centrality [31], betweenness centrality [32], VoteRank [33], K-core decomposition [34] and diffusion degree [35,36], etc., are efficient in identifying the most influential nodes based on network topology or integrated centrality measures. However, these methods are always trapped into suboptimal influence spread for the reason that the selected influential nodes in the seed set are always clustered neighbors, of which the influence spread tends to be overlapped easily.

Meta-heuristic algorithms, which mimic the cooperative behavior of biotic population or the evolutionary processes of physical phenomena, were validated to be effective and efficient in solving influence maximization problem. Jiang et al. [11] proposed an expected diffusion value (EDV) metric to measure the influence approximately of potential candidate nodes and adopted the simulated annealing algorithm to optimize and identify influential nodes. This is the first time meta-heuristic optimization algorithm was employed to solve influence maximization problem. The experimental results showed the algorithm runs faster by 2∼3 orders of magnitude than the simple greedy algorithm. Sankar et al. [37] proposed a bee algorithm for influence maximization by exploring the *waggle dance* behavior of bee colony and verified the performance of the proposed algorithm on the Twitter dataset. Gong et al. [12] formulated a local influence estimator to evaluate node influence within its two-hop areas and proposed a discrete particle swarm optimization (DPSO) to identify the top-k influential nodes. In addition, a memetic algorithm termed as CMA-IM [20] was put forwarded for influence maximization in community-based networks. Cui et al. [21] proposed the DDSE algorithm based on degree descending search evolutionary rules for the selection of targeted seed nodes. Experimental results demonstrated that it is a promising way to solve influence maximization problem using meta-heuristic algorithms. Besides the notable advantage that meta-heuristic algorithms can avoid generating Monte-Carlo simulations to estimate the expected influence spread of given node set, another one is that the running time of meta-heuristic algorithms for influence maximization is insensitive to different propagation probabilities, but not the greedy-based algorithms.

The major challenge faced by influence maximization algorithms is how to well balance the solution accuracy against reasonable time consumption and even the memory cost when tackling the influence maximization problem especially in large-scale networks. Therefore, developing effective and efficient algorithms for influence maximization is still filled with challenging research topics.

3. Preliminaries

3.1. Influence maximization problem

Definition 1. Let $G = (V, E)$ be a network, where V is the node set and E is the edge set of the network. Influence maximization problem aims to select targeted k $(1 \le k < |V|)$ influential nodes as seed set S such that the number of influenced nodes triggered by the seed set S, denoted as influence spread $\sigma(S)$, is maximum under a given propagation model.

$$S^* = \underset{S \subseteq V,\, |S|=k}{\arg\max}\, \sigma(S) \tag{1}$$

where S is a candidate seed set, $\sigma(S)$ is the expected number of influenced nodes that are triggered by S, and S^* is the best seed set that can maximize the spread of influence. As proved in [7], the influence maximization shown in Eq. (1) is an optimization problem.

3.2. Influence estimator model

Besides the subject of developing efficient methods to select a targeted seed set that can maximize the spread of influence, constructing effective mechanisms to estimate the expected influence spread accurately of a given node set is the second challenge of influence maximization. Ureña et al. [38] pointed out that it is hard to assess the agents' influence in real world social networks, especially when the whole network topology and related information is not given for granted.

Studies [39] on influence spreading dynamics in social network show that influence decays with one's friendship delimitation. More precisely, it was stated that the sum of the nearest neighbors' degree is a reliable local proxy for node's influence especially when the global network structure is unavailable and suggested to estimate the expected local influence spread within the two-hop area of a node. Based on the suggestion, the local influence estimator LIE can be formulated as in Eq. (2)[12].

$$LIE(S) = \sigma_0(S) + \sigma_1^*(S) + \tilde{\sigma}_2(S) \tag{2}$$

where $\sigma_0(S)$ is the size of seed set S, $\sigma_1^*(S)$ and $\tilde{\sigma}_2(S)$ are the expected influence spread of one-hop and two-hop area of set S, respectively. For the LIE of one-hop area and two-hop area can be expressed based the adjacency matrix of the nodes in S, then the LIE can be calculated according to Eq. (3)

$$\begin{aligned} LIE(S) &= k + \sigma_1^*(S) + \frac{\sigma_1^*(S)}{|N_S^{(1)} \backslash S|} \sum_{u \in N_S^{(2)} \backslash S} p_u^* d_u^* \\ &= k + \left(1 + \frac{1}{|N_S^{(1)} \backslash S|} \sum_{u \in N_S^{(2)} \backslash S} p_u^* d_u^*\right) \\ &\quad \times \sum_{i \in N_S^{(1)} \backslash S} \left(1 - \prod_{(i,j) \in E,\, j \in S} (1 - p_{i,j})\right) \end{aligned} \tag{3}$$

where $N_S^{(1)}$ and $N_S^{(2)}$ represent the one-hop and two-hop area of candidate set S, respectively. p_u^* is a small constant probability of a propagation cascade model. d_u^* is the number of edges of node u within $N_S^{(1)}$ and $N_S^{(2)}$.

Therefore the selection of k influential nodes is transformed into an optimization problem which aims at selecting a seed set to maximize the fitness value of Eq. (2). In this paper, we focus on providing an effective discrete frog-leaping algorithm to optimize the LIE function and explore the most influential nodes for influence maximization.

3.3. Influence propagation model

Based on the influence estimator, we employ the classical IC model [7] to simulate the spread of influence in given networks.

IC model is a probability model which mimics the spread process of information in social networks. In the IC model, each node has only two states, either *active* or *inactive*, and nodes can be allowed to switch from *inactive* to *active* ones, but not vice versa. Propagation probability p in the cascade model describes the tendency of inactive individuals to be affected by its adjacent active neighbors. Given an active node u at step t, it has only one chance to activate each of its adjacent inactive neighbors v and successes with a probability p_{uv}, which is associated with edge

$(u, v) \in E$. Whether the activation is succeed or not, u will no more attempt to activate v in the following steps. If node v is activated by u, then v will remain active and has one chance to activate each of its adjacent inactive neighbors in step $t+1$. The diffusion process terminates if no node is activated at step T and returns the influence spread $\sigma(S)$ comprising all of the active nodes.

4. Proposed method

As discussed above, the expected influence spread of given candidate nodes can be evaluated according to the local influence estimator, so optimization algorithms can be utilized to maximize the fitness value of the *LIE* function. Shuffled frog-leaping algorithm [40] is an advanced meta-heuristic algorithm, and its effectiveness on optimization problems has been validated in many studies [41,42]. Inspired by the efficient evolutionary mechanism based on swarm intelligence, we try to make further study on the algorithm and propose a discrete shuffled frog-leaping algorithm specially for influence maximization problem in this paper. In the following subsections, the basic memetic evolutionary ideology is introduced firstly, then discrete encoding mechanism and evolutionary rules are conceived for the virtual frog population based on the network topology characteristic, and then the framework of DSFLA for influence maximization is given consequently.

4.1. Memetic ideology and shuffled frog-leaping algorithm

Memetic algorithm (MA) is conceptualized to describe the population based meta-heuristic optimization algorithm. The term 'meme' in MA comes from [43], in which Dawkins considered the meme as a simple unit of intellectual or cultural information that survives long enough to be recognized and passed from mind to another. The components of a meme are called memetypes, such as an idea or information pattern, etc., which can cause someone to replace it or to repeat it to someone else.

As an important member of memetic algorithms, shuffled frog-leaping algorithm (SFLA), which was proposed by Eusuff et al. [40] for water distribution system design problem, is a population based cooperative search metaphor inspired by nature memetics. It has been validated to be simple and effective by many applications [41], in which Sarkheyli et al. reviewed the previous efforts from 89 researchers on SFLA and validated its effectiveness and robustness by comparing SFLA with widely used algorithms including particle swarm optimization (PSO), genetic algorithm (GA) and differential evolution (DE), etc., based on quantitative statistical results. To identify the main driving factors and define the contribution rate of the main factors to the net ecosystem exchange of carbon between the temperate forests and the atmosphere, Xue et al. [42] proposed a fuzzy rough set algorithm with binary shuffled frog leaping algorithm to optimize the objective function. To improve the performance of SFLA in dealing with multi-objective optimization problems, Luo et al. [44] proposed a modified evaluation strategy and a novel global optimal selection measure to diversify the flog population. Meanwhile, a multi-objective extremal optimization procedure was presented to enhance the evolution of the algorithm. Mao et al. [45] took advantage of the grouping concept of SFLA to improve the exploitation of PSO, and the experimental results on optimizing the power extraction problem proved that the modified SFLA outperforms PSO.

By idealizing the behavior rules of frogs, SFLA mimics a group of frogs leaping in a swamp, in which a number of stones locate at different positions, to find the stone that has the maximum amount of available food. Frogs in the group are partitioned into a number of memeplexes that are permitted to evolve independently. During the leaping process, frogs are allowed to communicate with each other so that they can improve their memes by learning from others' information. In each memeplex, the virtual frogs act as hosts of memes, and the meme carried by a frog is consisted of at least one memotype. The update of memotypes of a meme results in the improvement of the host frog's position approaching to the optimum.

As far as the influence maximization problem is concerned, the propagation process of node influence in the network is analogous to the evolution of memes in the frog population. Social individuals incline to profit from the sharing of promotional information and generalized behaviors of other interactive members and reshape their behavior as the influence propagates in the network. Therefore, the evolutionary mechanism of SFLA lends itself to be utilized to tackle influence maximization problem.

4.2. Discrete shuffled frog-leaping algorithm for influence maximization

4.2.1. Discrete encoding mechanism

To solve the influence maximization problem using SFLA, a reasonable encoding mechanism for the meme carried by frog individual is conceived based on the representation pattern of network topology structure. Given that each node in the network is identified by an unique nonnegative integer, then k different nodes can be drawn from the network to represent a meme individual, i.e., an arbitrary node represents one memetype of the meme. Fig. 1 shows the illustration of a virtual frog population.

4.2.2. Framework of DSFLA for influence maximization

According to the memetic ideology, the proposed framework of DSFLA is comprised of two phases including global exploration and local exploitation, which are detailed as follows. Meanwhile, novel discrete evolutionary rules are conceived based on network topology structure characteristics for the memes in the virtual frog population, which will be described in the following two evolutionary stages.

(1) Global exploration on the frog population.

In this first stage, global exploration for seed nodes is implemented on the m memeplexes, as shown in Fig. 2(A).

To generate the virtual frog population F, where $F = m \times n$, k different candidate nodes are drawn randomly from the network at a time to initialize the ith ($i = 1, \dots, F$) meme, denoted as $U(i) = (U_i^1, U_i^2, \dots, U_i^k)$. Consequently, the frog population can be formulated as $F = \{U(1), U(2), \dots, U(F)\}$. Meanwhile, the expected influence spread, denoted as $LIE(i)$, of meme $U(i)$ is calculated according to the local influence estimator as introduced in Section 3.2. Once the initialization is finished, sort the F memes in descending order according to their *LIE* fitness value and put the ordered combination $(U(i), f(i))$ into an array X, so that $U(1)$ represents the best frog with the maximal influence spread estimation value, $U(F)$ represents the worst frog, and the best frog of the entire population is recorded in P_X. Then the ordered array X are partitioned into m memeplexes $Y^1, Y^2, \dots, Y^m$ according to Eq. (4), so that each memeplex contains n ordered memes.

$$Y^i = [U(j)^i, f(j)^i | U(j)^i = U(i + m(j-1)), f(j)^i = f(i + m(j-1))] \tag{4}$$

where $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$.

A superiority of the partition mechanism is the influence spread estimation of each memeplex is uniformly distributed, so that the initial memeplexes located in the population are diversity.

(2) Local exploitation on each memeplex.

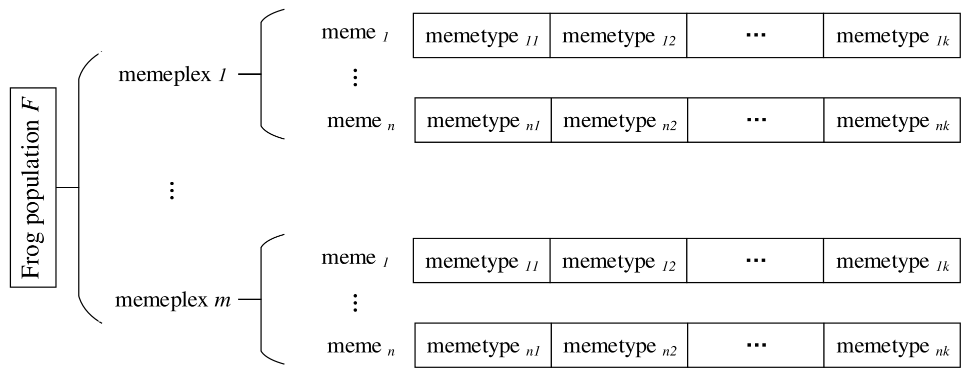

Fig. 1. Illustration of the virtual frog population F composed by $m$independent memeplexes. There are n memes in a memeplex, and each meme is consisted of k memetypes, i.e., k candidate nodes.

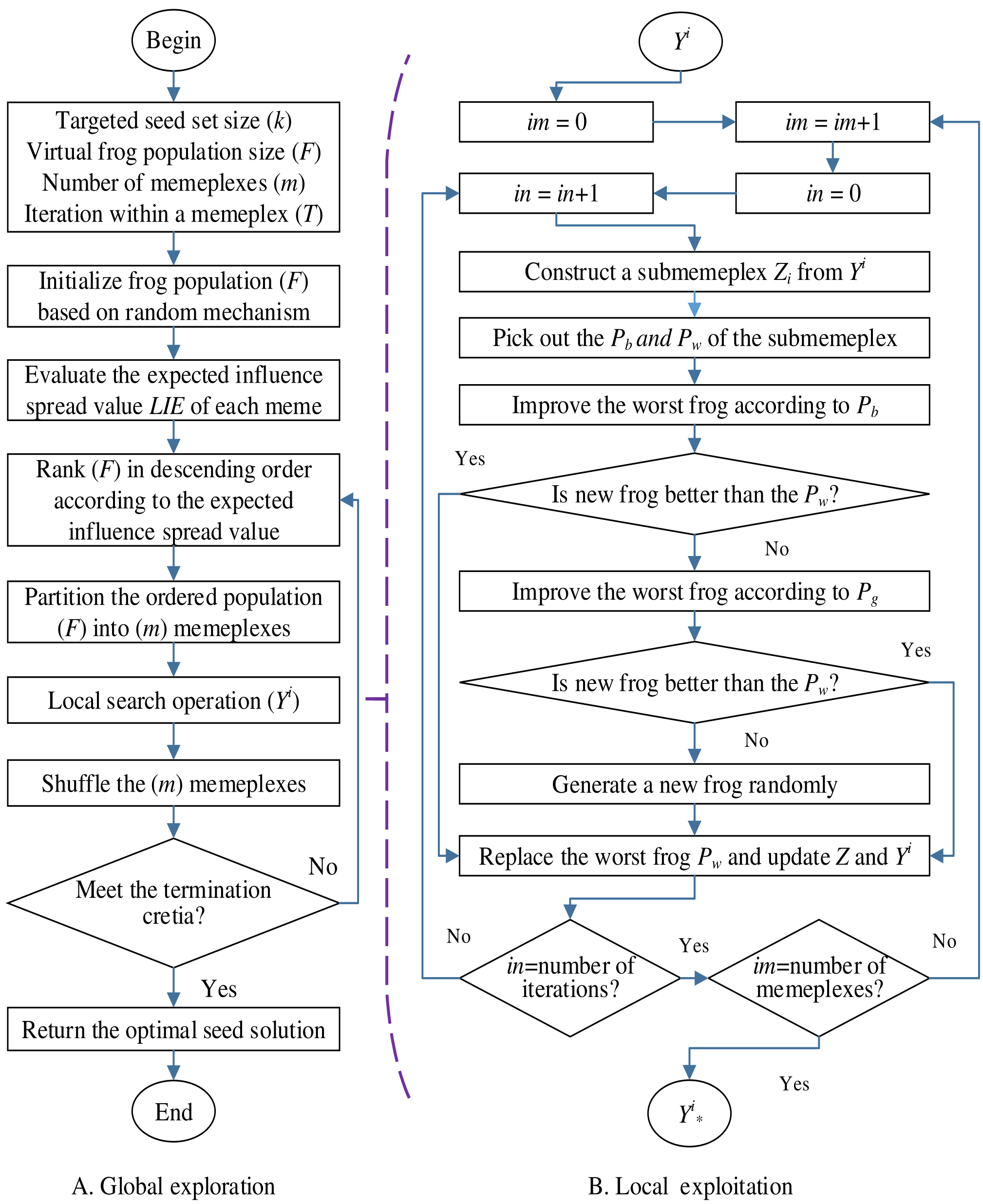

Fig. 2. The flowchart of the proposed DSFLA for influence maximization problem (where im is the index of memeplex, and in counts the number of local exploitation iterations on the current memeplex).

Following the partition operation, the local exploitation, as shown in Fig. 2(B), is adopted to improve the suboptimal meme of each memeplex independently according to a topology-based local exploitation strategy in this second stage. After the local exploitation is finished, shuffle the m memeplexes and put the ordered memes into X again for the next partition until the convergence criterion is satisfied.

As the target of discrete shuffled frog-leaping algorithm is to select k most influential nodes to maximize the expected influence spread, so the goal of each meme in the frog population is to find k most influential nodes that can maximize the fitness value of the local influence estimator. According to the memetic ideology, each meme in memeplex Y^i can improve its memotypes through an interactive local exploitation operation with the best meme within the same memeplex or with the global best solution P_X. In this paper, the property of network topology is further explored and a topology-based local exploitation strategy containing three schemes is presented to improve the suboptimal meme of each memeplex.

As stated in [40], it is not always a good idea to use the best frog to improve the worst one because the frogs' tendency would be to concentrate around that temporary best frog which may be a local optimal. Therefore, a subset of the memeplex called submemeplex, denoted as Z, is considered to be constructed from each memeplex according to the triangular probability distribution to improve the worst meme in the submemeplex. According to the construction strategy, frogs with larger influence spread estimation value will be given higher weights, and frogs with smaller influence spread estimation value will be given lower weights according to Eq. (5).

$$w_j = \frac{2(n+1-j)}{n(n+1)} \tag{5}$$

where n is the number of meme in each memeplex, and $j = 1, 2, \ldots, n$. Once the extraction is finished, the memes in submemeplex Z are sorted according to the influence spread estimation value, meanwhile, the best meme and the worst meme are recorded as P_b and P_w, respectively.

To improve the worst meme in the submemeplex Z_i ($i = 1, 2, \ldots, m$), a local degree-based replacement (*LDR*) method, as shown in Algorithm 1, is presented based on network topology for the improvement operation of the first two schemes in the local exploitation strategy. In Algorithm 1, the job of $Onehop(\cdot)$ is to get the direct neighbors of each memotype from the given meme. Function $Sort(\cdot)$ is utilized to rank the one-hop area nodes $N^{(1)}$ in decreasing order according to the degree centrality metric. In terms of the method, the one-hop area neighbors of each memotype from the best meme P_b (P_X) are obtained firstly, then k neighbors with the highest degree centrality from the k different one-hop areas are selected separately to make up a new meme to replace the worst meme P_w, meanwhile, it is essential to ensure that there is no overlapped nodes in the new meme. If the new memes generated based on P_b and P_X by the first two schemes are either infeasible or not better than the current meme P_w, then k nodes will be drawn randomly from the network to generate a new meme to replace P_w as well as its *LIE* value with the influence spread estimation value of the new generated meme.

As we can see that the local exploitation procedure plays an important role in promoting the proposed algorithm converges to global optimal solution. The worst meme in each submemeplex tends to return k more influential candidate nodes after given iterations. A foreseeable seed set with k most influential nodes will be returned by the proposed algorithm after predefined evolutionary generations.

Algorithm 1: Local degree-based replacement method.

```
Input: P_b of submemeplex Z_i or P_X of the entire population.
new_meme ← Φ
for each memetype ∈ P_b (P_X) do
  N^(1) ← Onehop(memetype)
  SN^(1) ← Sort(N^(1))
   for each node ∈ SN^(1) do
      if node ∉ new_meme then
         new_meme ← new_meme ∪ {node}
         break
      end if
   end for
end for
return new_meme
```

4.3. Time complexity of DSFLA

The asymptotic time complexity of the proposed DSFLA is analyzed in this subsection to check whether the algorithm is effective in selecting the targeted seed nodes for influence maximization.

Proposition 4.1. *Given the proposed framework of DSFLA for influence maximization problem, the frog population initialization, memeplex partition and Shuffling the m memeplexes have a complexity of* $O(mn)$*, respectively.*

Proof. If the virtual frog population F is divided into m memeplexes, and there are n memes in each memeplex, then it needs $O(n)$ basic operations to initialize one meme based on random candidate node generation, therefore, the asymptotic time complexity of the initialization is $O(mn)$. Meanwhile, according to the partitioning rules, the fast way to partition the F ordered frogs into m memeplexes needs $O(mn)$ basic operations. In addition, after the independent local improvement evolution, all the memes need to be shuffled into the population array again for the next generation, for there are m memeplexes and n memes in each memeplex, therefore, the time complexity of the shuffling operation is $O(mn)$. □

Proposition 4.2. *Evaluating the expected influence spread LIE of a meme needs* $O(k\bar{D})$ *basic operations.*

Proof. Let $\bar{D}$ is the average node degree of the network. When evaluating the expected local influence spread of a given candidate node, we can obtain the number of its direct neighbors, then it requires $O(\bar{D})$ basic operations to obtain the number of edges between its direct neighbors and two-hop area neighbors. Therefore, it requires $O(k\bar{D})$ basic operations to evaluate the expected influence spread of the candidate seed set S. □

Proposition 4.3. *Ranking the F frogs requires* $O(mn \cdot log(mn))$*.*

Proof. There are $m \times n$ frogs in the virtual population, so the worst case to rank the $m \times n$ frogs according to the fitness value based on the merging sort method is $O(mn \log(mn))$. □

Proposition 4.4. *The asymptotic time complexity of the local exploitation is* $O(mn + mk\bar{D})$*.*

Proof. Firstly, it needs $O(n)$ to calculate a weight vector for the n memes according to Eq. (5). Constructing a submemeplex requires $O(n)$ basic operations, the improve the worst meme P_w of each memeplex needs $O(k\bar{D})$. Therefore, the total asymptotic time complexity is $O(mn + mk\bar{D})$. □

Table 1
Statistical characteristics of the six social networks. $|V|$ and $|E|$ represent the number of nodes and edges, respectively. $\langle k \rangle$ is the average node degree, $\bar{d}$ is the average shortest path distance, C represents the average clustering coefficient, AC represents the assortativity coefficient.

Networks	$\|V\|$	$\|E\|$	$\langle k \rangle$	$\bar{d}$	C	AC	*Type*
AstroPh	18772	198110	21.107	4.194	0.677	0.205	Undirected
CondMat	23133	186936	16.162	5.352	0.055	0.135	Undirected
Slashdot	77360	905468	23.409	4.024	0.087	−0.046	Undirected
Epinions	75879	508837	13.412	4.755	0.261	−0.041	Directed
Eu-Email	265214	420045	3.168	4.206	0.456	−0.210	Directed
Stanford	281903	2312497	16.406	6.824	0.598	−0.122	Directed

Proposition 4.5. *The asymptotic time complexity of the proposed DSFLA for influence maximization problem is* $O(mnk\bar{D} + (mnlog(mn) + mk\bar{D})g_{max})$.

Proof. Based on the propositions given above, let the other operations need one unit cost separately and g_{max} is the maximal number of evolutionary generations of DSFLA, the upper bound computational complexity of DSFLA is $O(mnk\bar{D} + (mnlog(mn) + mk\bar{D})g_{max})$. □

Compared with the time complexity $O(kNMR)$ of the simple greedy algorithm (where the size of N and M is much larger than the size of m and n), we can see that the proposed DSFLA algorithm is more efficient.

5. Experiments and statistical tests

5.1. Datasets and baseline algorithms

To validate the performance of the proposed DSFLA on influence maximization problem, experiments are carried out on six real-world social networks, as shown in Table 1. AstroPh and CondMat [46] are two undirected collaboration networks which cover scientific collaborations between authors of papers submitted to Arxiv Astro Physics and Condensed Matter, respectively. Slashdot [47] is a technology-related news social network known for its specific user community, and it is treated as an undirected network. Epinions [48] is a who-trust-whom online social network of a general consumer review site Epinions.com. Members of the site can decide whether to "trust" each other, and the trust relationships are represented by directed edges. Eu-Email [46] network is generated from a large European research institution, in which each node corresponds to an email address and each sent or received email message corresponds to an interactive edge. Stanford [47] is a large-scale web graph extracted from Stanford University (stanford.edu), in which nodes represent pages from the web and directed edges represent hyperlinks between the nodes. The node degree distribution of the six networks is given in Fig. 3.

The experiments on the performance of DSFLA consist of two separate phases. In the first phase, experiments on parameters setting strategy for DSFLA are carried out, and an optimal set of combinational parameters setting is selected so that the proposed DSFLA can converge to global optimization efficiently. Then, performance of DSFLA on influence spread is validated by comparing with four other state-of-the-art algorithms under IC model at propagation probability $p = 0.01$ on the six social networks. Statistical hypothesis tests according to two significance levels are also performed based on the experimental results.

• **CELF** [8] (Cost-Effective Lazy Forward) is a notable greedy-based algorithm with a "lazy-forward" strategy by exploiting the submodularity property. CELF runs k rounds to select the targeted k seed nodes. At each round, the algorithm performs at least 10,000 Monte-Carlo simulations under the given IC model to estimate the marginal gain of each node in the network. Then the node with the largest marginal gain is to be selected as the current best seed node and the total influence spread accumulates by the seed node's marginal gain.

• **DPSO**[12] (Discrete Particle Swarm Optimization) is an effective meta-heuristic algorithm that selects the candidate node set with the best *LIE* fitness value as the targeted k seed nodes after a given number of optimizing iterations. Then the average influence spread is calculated after simulating the spreading process of the seed set in the network by a predefined number of times.

• **DDSE**[21] (Degree-Descending Search Evolution) is a novel memetic algorithm originated from Differential Evolutionary (DE) algorithm based on degree-descending search strategy for influence maximization, and the calculation of influence spread is similar to DPSO.

• **SSA**[28] (Stop-and-Stare Algorithm) is an optimal sampling framework based on reverse influence sampling ideology. The algorithm selects the top k nodes that appear the most times in the generated subgraphs as the optimal seed set, then the average influence spread is calculated after simulating the spreading process of the seed set in the network by a predefined number of times.

5.2. Parameters setting of DSFLA

Parameters including the number of frog individual F, the number of memeplexes m as well as the related number of memes n within each memeplex need to be settled firstly for the proposed DSFLA. In addition, it needs to determine the evolution iterations *Iter* of the local exploitation, the submemeplex size q, and the maximum generation number g_{max} allowed for the algorithm. Conventional numerical test or set value to the parameters empirically is inefficient, and it is hard to conduct all the experiments in enumerations of all possible scenarios. Take the parameter F for example, smaller F always indicates DSFLA cannot explore sufficiently in the solution space and tends to lead the algorithm to local optimal solution especially in large-scale networks, while larger F may result in extra and verbose computation. As emphasized by Eusuff et al. [40], parameters setting strategy is critical to SFLA's performance, but no clear theoretical basis is available to dictate parameters setting.

In this paper, the orthogonal experimental design method is employed to optimize the parameters setting strategy of DSFLA. The method provides a highly efficient way of dealing with multi-factor experiments and screening optimum levels by using an orthogonal design table. To make an orthogonal design table, reasonable and representative levels of factors need to be determined at first according to theories or experiments. We mainly choose four factors including F, m, q and *Iter* and set three alternative scenarios for each factor respectively to represent all the level groups of the experimental factors. Then an orthogonal design table $L_9[3^4]$, as shown in Table 2, is constructed and the orthogonal test is implemented on the CondMat network in this paper. From the results shown in Table 2, we can see that DSFLA achieves the best and robust performance under the third parameter setting pattern, where the average expected local influence estimation is $AveLIE = 105.147,6$ and the standard difference is $SD = 0.417,5$ when the targeted seed size is $k = 100$. Therefore, the following parameter setting pattern where $F = 100$, $m = 20$, $n = 5$, $q = 3 * n/4$ and $Iter = 30$ is fixed for DSFLA to solve the influence maximization on the six networks.

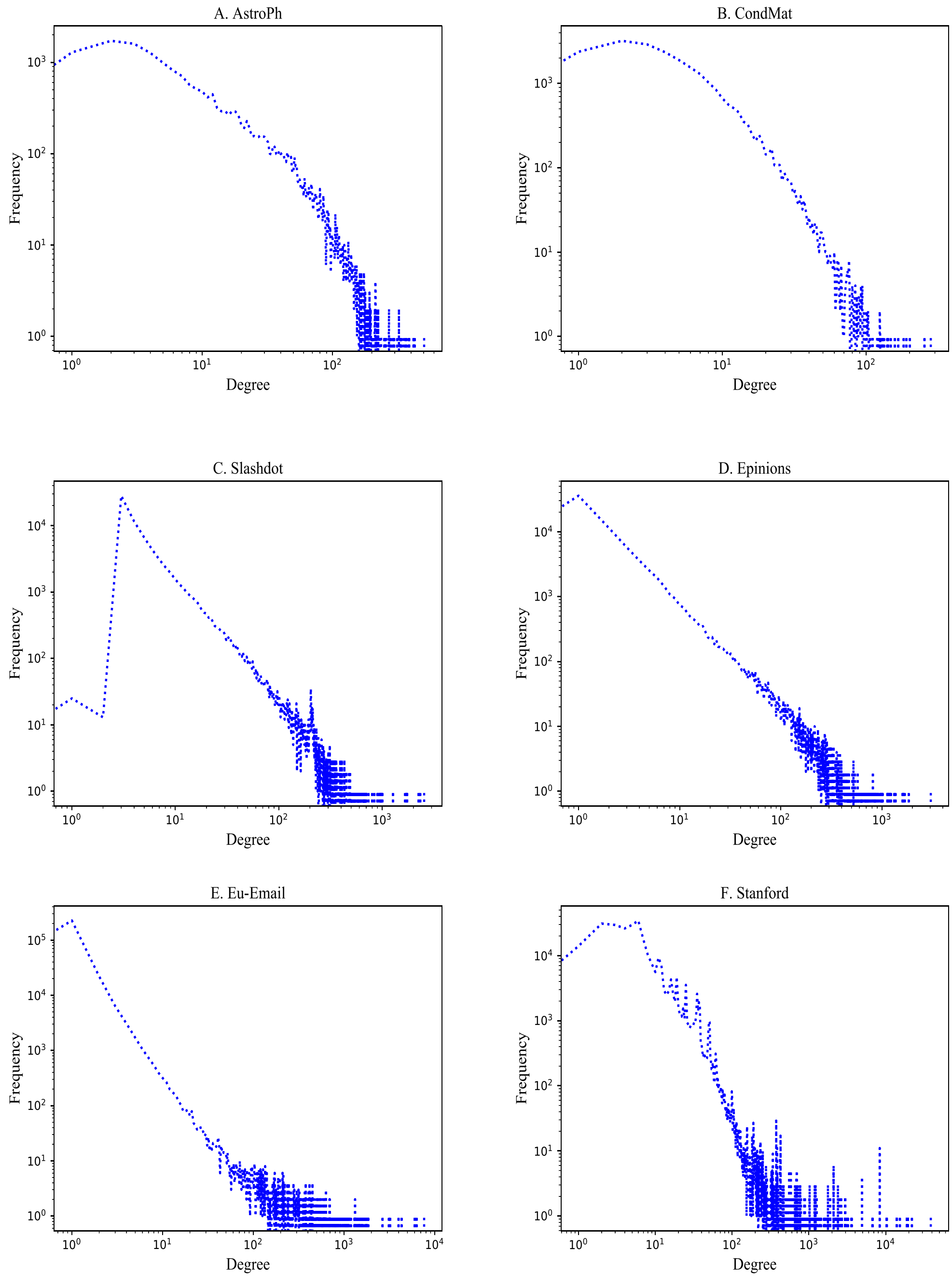

Fig. 3. Node degree distribution of the six different social networks.

Table 2
The orthogonal tests on the four key parameters of DSFLA based on CondMat network (*AveLIE* is the average expected local influence estimation, *SD* is the standard difference of *AveLIE* after DSFLA runs 50 times independently).

Tests	Orthogonal test				Running time(s)	*AveLIE*	*SD*
	F	*m*	*q*	*Iter*			
*test*1	100	5	n/3	10	50.890,0	102.976,8	2.517,2
*test*2	100	10	2∗n/3	20	254.425,0	104.214,4	0.908,7
*test*3	100	20	3∗n/4	30	129.748,0	105.147,6	0.417,5
*test*4	200	5	2∗n/3	30	108.416,0	104.272,0	0.830,4
*test*5	200	10	3∗n/4	10	140.353,3	103.908,8	0.949,9
*test*6	200	20	n/3	20	506.200,0	104.663,8	0.723,6
*test*7	300	5	3∗n/4	20	271.968,0	104.219,1	0.976,5
*test*8	300	10	n/3	30	354.496,5	103.774,0	1.112,1
*test*9	300	20	2∗n/3	10	252.612,0	104.421,5	0.793,2

5.3. Influence spread comparison

It is important to note that all related parameters value are fixed according to the suggestions in the original literature when we implement the procedures of the four baseline algorithms. We run the Monte-Carlo simulation 10,000 times for CELF to estimate the marginal influence spread of each node. The maximal evolutionary generation g_{max} for DSFLA, DPSO and DDSE is set to 100 respectively, and the simulation times for the three algorithms is set to 1,000 separately to obtain the average influence spread. The learning factors c_1 and c_2 in PSO are set to 2, and the inertia weight ω is set to 0.8. The probabilities of mutation, crossover and diversity operations in DDSE are set to 0.1, 0.4 and 0.6, respectively. For the SSA algorithm, parameters ϵ and δ are set to 0.1 and 0.01, respectively.

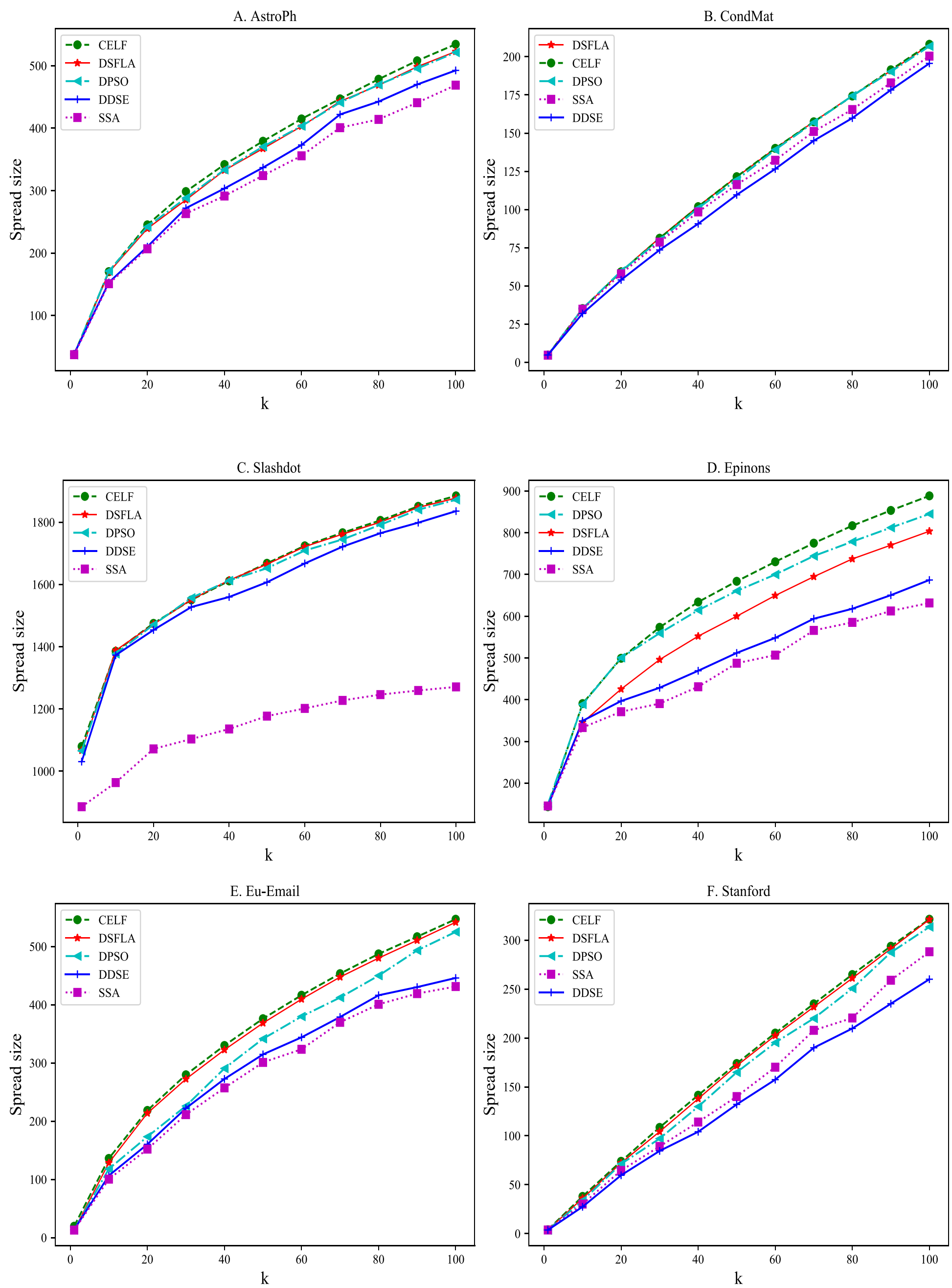

Fig. 4. Comparison on influence spread of the five algorithms under IC model at propagation probability $p = 0.01$ on the six social networks.

Fig. 4 shows the performance of DSFLA and other four baseline algorithms on influence spread under IC model at propagation probability $p = 0.01$ on the six networks. As shown in Fig. 4(A)-(F), DSFLA achieves satisfying influence spread at given seed size in the six large-scale networks, and the smooth marginal gains with the increment of targeted seed size show that DSFLA is robust in identifying the targeted seed nodes for influence maximization. Comparing with the four other state-of-the-art algorithms, DSFLA achieves comparable influence spread to CELF and even better solutions than CELF on the CondMat network, as shown in Fig. 4(B). Meanwhile, DSFLA outperforms DPSO, DDSE and SSA under all of the scenarios except on the Epinions network, as shown in Fig. 4(D). In other words, the proposed DSFLA is effective in identifying influential nodes due to its memetic evolutionary rules.

As to the DPSO, it achieves less influence spread compared to CELF and DSFLA in most scenarios except the one in Fig. 4(D). The evolutionary rules of DPSO are efficient, but the local search strategy, which always terminates once the local search operation fails to find a better influential node to replace the current candidate node, employed in DPSO tends to lead the algorithm to be trapped into local optimal solution easily. As shown in Fig. 4, DDSE acts itself as the worst one among the three meta-heuristic algorithms, this is because the algorithm suffers from the problem of rough estimation of node's influence spread, which estimates the expected diffusion value of a node within its one-hop area. Among the five algorithms, SSA performs as the worst one in identifying the most influential nodes for influence maximization. As shown in Fig. 4(A), (B) and (D)∼(F), SSA is effective when the targeted seed set size is small, such as $k \leq 10$, however, the algorithm tends to fail to identify the most influential nodes with the increment of the targeted seed set size k, especially in Fig. 4(C) and (D).

Table 3
Statistical results of the multiple-problem Wilcoxon's test for the five algorithms at $\alpha = 0.05$ and $\alpha = 0.1$ significance levels.

DSFLA *vs*	k	N+	N-	Z	p-value	Adjusted p-value		$\alpha = 0.1$	$\alpha = 0.05$
						Holm	Hochberg		
CELF	1	1	5	−1.363	0.173	1	0.753	NO	NO
	10	1	5	−1.363	0.173	1	0.753	NO	NO
	20	1	5	−1.992	0.046	0.128	0.128	YES	YES
	30	2	4	−1.572	0.116	0.232	0.232	NO	NO
	40	2	4	−1.572	0.116	0.232	0.232	NO	NO
	50	0	6	−2.201	0.028	0.112	0.056	YES	YES
	60	1	5	−1.572	0.116	0.232	0.232	NO	NO
	70	1	5	−1.992	0.046	0.128	0.128	YES	YES
	80	1	5	−1.992	0.046	0.128	0.128	YES	YES
	90	0	6	−2.201	0.028	0.112	0.056	YES	YES
	100	0	6	−2.201	0.028	0.112	0.056	YES	YES
DPSO	1	3	3	−0.674	0.500	1	0.753	NO	NO
	10	3	3	−0.314	0.753	1	0.753	NO	NO
	20	4	2	−0.105	0.917	0.917	0.917	NO	NO
	30	5	1	−0.943	0.345	0.345	0.345	NO	NO
	40	5	1	−0.943	0.345	0.345	0.345	NO	NO
	50	4	2	−0.524	0.600	0.6	0.6	NO	NO
	60	5	1	−0.943	0.345	0.345	0.345	NO	NO
	70	5	1	−0.943	0.345	0.345	0.345	NO	NO
	80	5	1	−0.943	0.345	0.345	0.345	NO	NO
	90	5	1	−0.943	0.345	0.345	0.345	NO	NO
	100	5	1	−0.943	0.345	0.345	0.345	NO	NO
DDSE	1	2	4	−0.524	0.600	1	0.753	NO	NO
	10	6	0	−2.201	0.028	0.112	0.112	YES	YES
	20	6	0	−2.201	0.028	0.112	0.084	YES	YES
	30	6	0	−2.201	0.028	0.112	0.084	YES	YES
	40	6	0	−2.201	0.028	0.112	0.084	YES	YES
	50	6	0	−2.201	0.028	0.112	0.056	YES	YES
	60	6	0	−2.201	0.028	0.112	0.084	YES	YES
	70	6	0	−2.201	0.028	0.112	0.084	YES	YES
	80	6	0	−2.201	0.028	0.112	0.084	YES	YES
	90	6	0	−2.201	0.028	0.112	0.056	YES	YES
	100	6	0	−2.201	0.028	0.112	0.056	YES	YES
SSA	1	3	3	−0.314	0.753	1	0.753	NO	NO
	10	5	1	−1.992	0.046	0.138	0.138	YES	YES
	20	6	0	−2.201	0.028	0.112	0.084	YES	YES
	30	6	0	−2.201	0.028	0.112	0.084	YES	YES
	40	6	0	−2.201	0.028	0.112	0.084	YES	YES
	50	6	0	−2.201	0.028	0.112	0.056	YES	YES
	60	6	0	−2.201	0.028	0.112	0.084	YES	YES
	70	6	0	−2.201	0.028	0.112	0.084	YES	YES
	80	6	0	−2.201	0.028	0.112	0.084	YES	YES
	90	6	0	−2.201	0.028	0.112	0.056	YES	YES
	100	6	0	−2.201	0.028	0.112	0.056	YES	YES

5.4. Running time comparison

To show the efficiency of DSFLA in identifying influential nodes for influence maximization, comparison on running time of the five algorithms at targeted seed set size $k = 100$ on the six networks is given in Fig. 5.

The bar charts in Fig. 5 show that both SSA and DDSE need merely tens of seconds to identify the targeted seed nodes on the six large-scale networks. However, in consideration of the influence spread shown in Fig. 4, SSA and DDSE tend to be the less effective than other three algorithms. Conversely, the bar charts illustrate that the greedy-based CELF is the most time consuming one, where it even needs 31 h to select the targeted seed set in the network Slashdot, though it performs as the most effective method compared with other algorithms. Comparing with DPSO and CELF, the proposed DSFLA performs more efficient at identifying influential nodes and can be scalable to large-scale networks, as shown in Fig. 5, where the time consumed by DSFLA is only half of DPSO in the largest Stanford network and almost 2432 times faster than CELF on the Slashdot network.

5.5. Statistical tests

To verify the effectiveness of DSFLA independently, we also carry out rigorous statistical hypothesis tests in terms of quartile statistics to check whether there is a high level of statistical significance in the results of the five algorithms on the six networks. In each of the networks, 11 scenarios ($k = 1, 10, \dots, 100$) are considered as independent problems, moreover, the parameter-free hypothesis tests on each k scenario on the six networks are carried out separately. The multiple-problem Wilcoxon's tests [49] are performed to check the behaviors of the five algorithms, in which Holm procedure and Hochberg procedure are used as post-hoc procedures.

Table 3 summarizes the statistical analysis results by taking the proposed DSFLA as the baseline. According to the statistical results, we can see that DSFLA is significantly better than DDSE and SSA, but there is no significant difference between DSFLA and DPSO.

The experimental comparison on influence spread and the independent statistical tests show that the proposed DSFLA is an advanced and effective algorithm for influence maximization problem. In general, there are always many parameters and factors need to be adjusted in meta-heuristic algorithms, reasonable parameters setting strategy involves the convergence ability and the performance of these algorithms. We can see that shuffling and partitioning the frog population into m memeplexes alternatively contributes to diversify the solutions, and the local exploitation strategy on the worst meme within each submemeplex is able to exploit the most influential nodes with the help

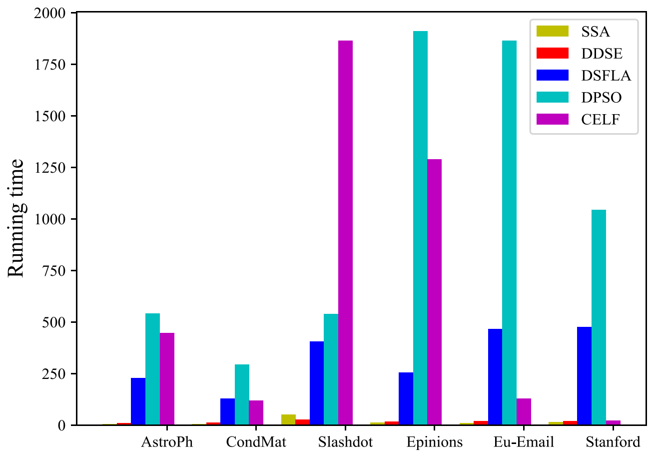

Fig. 5. Comparison on running time of the five algorithms at the targeted seed set size $k = 100$ on the six networks (where the running time of CELF in the six networks is measured in minutes, while the other four are measure in seconds).

of collective evolution. Comparing with DDSE, it shows us that an effective influence estimator makes it easier to identify influential nodes accurately. Meanwhile, the comparison on running time shows that time complexity is a major factor in evaluating an algorithm whether it can be scalable to the influence maximization problem in large-scale social networks. Therefore, developing meta-heuristic algorithms based on memetic ideology is a promising way to solve influence maximization problem in large-scale networks.

6. Conclusions and future works

The shuffled frog-leaping algorithm which combines deterministic and random search strategies shows excellent performance on various complex optimization problems. In this paper, a discrete shuffled frog-leaping algorithm is proposed specially to identify influential nodes for influence maximization. In the proposed framework, discrete encoding mechanism and evolutionary rules are conceived based on network topology, and a local degree-based replacement strategy is presented to cooperate with the local exploitation to improve the suboptimal meme of each memeplex. Meanwhile, the orthogonal experimental design method is employed to optimize the parameters setting strategy of DSFLA so that the algorithm evolutes effectively.

The experimental results on various cases demonstrate that the proposed method can successfully identify the influential nodes in networks. Compared with the methods that identifies the influential nodes based on Monte-Carlo simulations or reversed influence sampling ideology, the evolutionary mechanisms which take the local spreading of influence in the network into consideration are more reasonable and efficient. The excellent performance and the independent statistical tests of the proposed algorithm supports the claim that meta-heuristics based on swarm intelligence are promising tools to solve influence maximization problem in large-scale networks. As a future work, developing effective influence spread estimators and more advanced evolutionary rules that are scalable to large-scale networks is one of the main focus of our further research on influence maximization problem.

Acknowledgments

This work is supported by the National Natural Science Foundations of China (Grant No. 21503101 and No. 61702240) and the CERNET Innovation Project, China (NGII20170422).

References

[1] R. Ureña, G. Kou, Y. Dong, F. Chiclana, E. Herrera-Viedma, A review on trust propagation and opinion dynamics in social networks and group decision making frameworks, Inform. Sci. 478 (2019) 461–475, http://dx.doi.org/10.1016/j.ins.2018.11.037.

[2] S. Peng, Y. Zhou, L. Cao, S. Yu, J. Niu, W. Jia, Influence analysis in social networks: A survey, J. Netw. Comput. Appl. 106 (2018) 17–32.

[3] J.J. Brown, P.H. Reingen, Social ties and word-of-mouth referral behavior, J. Consum. Res. 14 (3) (1987) 350–362.

[4] P. Domingos, M. Richardson, Mining the network value of customers, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 57–66.

[5] T. Zhu, B. Wang, B. Wu, C. Zhu, Maximizing the spread of influence ranking in social networks, Inform. Sci. 278 (2014) 535–544.

[6] J.R. Lee, C.W. Chung, A query approach for influence maximization on specific users in social networks, IEEE Trans. Knowl. Data Eng. 27 (2) (2015) 340–353.

[7] D. Kempe, J. Kleinberg, Maximizing the spread of influence through a social network, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137–146.

[8] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. Vanbriesen, N. Glance, Cost-effective outbreak detection in networks, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 420–429.

[9] S. Kundu, S.K. Pal, Deprecation based greedy strategy for target set selection in large scale social networks, Inform. Sci. 316 (C) (2015) 107–122.

[10] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 199–208, http://dx.doi.org/10.1145/1557019.1557047.

[11] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, K. Xie, Simulated annealing based influence maximization in social networks, in: AAAI Conference on Artificial Intelligence, 2011, pp. 127–132.

[12] M. Gong, J. Yan, B. Shen, L. Ma, Q. Cai, Influence maximization in social networks based on discrete particle swarm optimization, Inform. Sci. 367–368 (C) (2016) 600–614.

[13] M. Kimura, K. Saito, Tractable models for information diffusion in social networks, in: J. Fürnkranz, T. Scheffer, M. Spiliopoulou (Eds.), Knowledge Discovery in Databases: PKDD 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 259–271.

[14] S. Kim, D. Kim, J. Oh, J.H. Hwang, W.S. Han, W. Chen, H. Yu, Scalable and parallelizable influence maximization with random walk ranking and rank merge pruning, Inform. Sci. (2017).

[15] T. Cao, X. Wu, S. Wang, X. Hu, OASNET: An optimal allocation approach to influence maximization in modular social networks, in: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC '10, ACM, New York, NY, USA, 2010, pp. 1088–1094, http://dx.doi.org/10.1145/1774088.1774314.

[16] J. Shang, S. Zhou, X. Li, L. Liu, H. Wu, CoFIM: A Community-based framework for influence maximization on large-scale networks, Knowl.-Based Syst. 117 (2017) 88–100.

[17] E.M. Golafshani, Introduction of biogeography-based programming as a new algorithm for solving problems, Appl. Math. Comput. 270 (2015) 1–12, http://dx.doi.org/10.1016/j.amc.2015.08.026, http://www.sciencedirect.com/science/article/pii/S0096300315010814.

[18] J. Wei, R. Zhang, Z. Yu, R. Hu, J. Tang, C. Gui, Y. Yuan, A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanisms for feature selection, Appl. Soft Comput. 58 (2017) 176–192, http://www.sciencedirect.com/science/article/pii/S1568494617302430.

[19] I. Fister, S. Rauter, X.-S. Yang, K. Ljubič, I. Fister, Planning the sports training sessions with the bat algorithm, Neurocomputing 149 (2015) 993–1002, http://www.sciencedirect.com/science/article/pii/S0925231214009710.

[20] M. Gong, C. Song, C. Duan, L. Ma, An efficient memetic algorithm for influence maximization in social networks, IEEE Comput. Intell. Mag. 11 (3) (2016) 22–33.

[21] L. Cui, H. Hu, S. Yu, Q. Yan, Z. Ming, Z. Wen, N. Lu, DDSE: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks, J. Netw. Comput. Appl. 103 (2018) 119–130.

[22] C. Gui, R. Zhang, R. Hu, G. Huang, J. Wei, Overlapping communities detection based on spectral analysis of line graphs, Physica A 498 (2018) 50–65.

[23] J. Shang, H. Wu, S. Zhou, J. Zhong, Y. Feng, B. Qiang, IMPC: Influence maximization based on multi-neighbor potential in community networks, Physica A 512 (2018) 1085–1103.

[24] F. Lu, W. Zhang, L. Shao, X. Jiang, P. Xu, H. Jin, Scalable influence maximization under independent cascade model, J. Netw. Comput. Appl. 86 (2016) 15–23.

[25] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, Philadelhia, PA, USA, 2014, pp. 946–957.
[26] Y. Tang, X. Xiao, Y. Shi, Influence maximization: near-optimal time complexity meets practical efficiency, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD '14, 2014, pp. 75–86.
[27] E. Cohen, D. Delling, T. Pajor, R.F. Werneck, Sketch-based influence maximization and computation: Scaling up with guarantees, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, ACM, New York, NY, USA, 2014, pp. 629–638, http://dx.doi.org/10.1145/2661829.2662077.
[28] H.T. Nguyen, M.T. Thai, T.N. Dinh, Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks, in: Proceedings of the 2016 International Conference on Management of Data, SIGMOD '16, ACM, New York, NY, USA, 2016, pp. 695–710, http://dx.doi.org/10.1145/2882903.2915207.
[29] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time: a martingale approach, in: ACM SIGMOD International Conference on Management of Data, 2015, pp. 1539–1554.
[30] J. Tang, R. Zhang, Y. Yao, Z. Zhao, P. Wang, H. Li, J. Yuan, Maximizing the spread of influence via the collective intelligence of discrete bat algorithm, Knowl.-Based Syst. 160 (2018) 88–103, http://www.sciencedirect.com/science/article/pii/S0950705118303423.
[31] R. Cohen, K. Erez, D. ben Avraham, S. Havlin, Breakdown of the internet under intentional attack, Phys. Rev. Lett. 86 (2001) 3682–3685.
[32] M.J. Newman, A measure of betweenness centrality based on random walks, Social Networks 27 (1) (2005) 39–54.
[33] J.-X. Zhang, D.-B. Chen, Q. Dong, Z.-D. Zhao, Identifying a set of influential spreaders in complex networks, Sci. Rep. 6 (2016) 27823, http://dx.doi.org/10.1038/srep27823.
[34] M.A. Al-garadi, K.D. Varathan, S.D. Ravana, Identification of influential spreaders in online social networks using interaction weighted K-core decomposition method, Physica A 468 (2017) 278–288.
[35] S. Kundu, C.A. Murthy, S.K. Pal, A new centrality measure for influence maximization in social networks, in: Proceedings of the 4th International Conference on Pattern Recognition and Machine Intelligence, PReMI'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 242–247.
[36] S.K. Pal, S. Kundu, C.A. Murthy, Centrality measures, upper bound, and influence maximization in large scale directed social networks, Fundam. Inf. 130 (3) (2014) 317–342.
[37] C.P. Sankar, S. Asharaf, K.S. Kumar, Learning from bees: An approach for influence maximization on viral Campaigns, PLoS One 11 (12) (2016) e0168125.
[38] R. Ureña, F. Chiclana, G. Melançon, E. Herrera-Viedma, A social network based approach for consensus achievement in multiperson decision making, Inf. Fusion 47 (2019) 72–87, http://dx.doi.org/10.1016/j.inffus.2018.07.006.
[39] S. Pei, L. Muchnik, J.S. Andrade, Z. Zheng, H.A. Makse, Searching for superspreaders of information in real-world social media, Sci. Rep. 4 (2014) 5547.
[40] M. Eusuff, K. Lansey, F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Eng. Optim. 38 (2) (2006) 129–154.
[41] A. Sarkheyli, A.M. Zain, S. Sharif, The role of basic, modified and hybrid shuffled frog leaping algorithm on optimization problems: a review, Soft Comput. 19 (7) (2015) 2011–2038.
[42] Y. Xue, Y. Chen, Y. Hu, H. Chen, Fuzzy Rough set algorithm with binary shuffled frog-leaping (BSFL-frsa): An innovative approach for identifying main drivers of carbon exchange in temperate deciduous forests, Ecol. Indicators 83 (2017) 41–52.
[43] R. Dawkins, The selfish gene, Q. Rev. Biol. 110 (466) (1976) 781–804, (24).
[44] J. Luo, Y. Yang, Q. Liu, X. Li, M. Chen, K. Gao, A new hybrid memetic multi-objective optimization algorithm for multi-objective optimization, Inform. Sci. 448–449 (2018) 164–186.
[45] M. Mao, L. Zhang, P. Duan, Q. Duan, M. Yang, Grid-connected modular PV-converter system with shuffled frog leaping algorithm based DMPPT controller, Energy 143 (2018) 181–190, http://dx.doi.org/10.1016/j.energy.2017.10.099.
[46] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Trans. Knowl. Discov. Data 1 (1) (2007) http://doi.acm.org/10.1145/1217299.1217301.
[47] J. Leskovec, K.J. Lang, A. Dasgupta, M.W. Mahoney, Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters, Internet Math. 6 (1) (2009) 29–123.
[48] M. Richardson, Trust management for the semantic web, in: International Semantic Web Conference, 2003, https://ci.nii.ac.jp/naid/10016795094/en/.
[49] S. Garcí a, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization, J. Heuristics 15 (6) (2009) 617–644.