Physica A 513 (2019) 477–496

Contents lists available at ScienceDirect

Physica A

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

Identification of top-k influential nodes based on enhanced discrete particle swarm optimization for influence maximization

Jianxin Tang [a,b], Ruisheng Zhang [a,*], Yabing Yao [a], Fan Yang [a], Zhili Zhao [a], Rongjing Hu [a], Yongna Yuan [a]

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

HIGHLIGHTS

- We make further exploration on discrete PSO for influence maximization problem.
- We design a network topology-based local search to enhance the exploitation of DPSO.
- Our algorithm achieves comparable results to greedy algorithm and the CELF algorithm.
- The proposed algorithm provides steady performance on identifying influential nodes.

ARTICLE INFO

Article history:
Received 30 October 2017
Received in revised form 7 August 2018
Available online 7 September 2018

Keywords:
Social networks
Influence maximization
Metaheuristic
Discrete particle swarm optimization
Local search strategy

ABSTRACT

Influence maximization aims to select a subset of top-k influential nodes to maximize the influence propagation, and it remains an open research topic of viral marketing and social network analysis. Submodularity-based methods including greedy algorithm can provide solutions with performance guarantee, but the time complexity is unbearable especially in large-scale networks. Meanwhile, conventional centrality-based measures cannot provide steady performance for multiple influential nodes identification. In this paper, we propose an improved discrete particle swarm optimization with an enhanced network topology-based strategy for influence maximization. According to the strategy, the k influential nodes in a temporary optimal seed set are recombined firstly in ascending order by degree metric to let the nodes with lower degree centrality exploit more influential neighbors preferentially. Secondly, a local greedy strategy is applied to replace the current node with the most influential node from the direct neighbor set of each node from the temporary seed set. The experimental results conducted in six social networks under independent cascade model show that the proposed algorithm outperforms typical centrality-based heuristics, and achieves comparable results to greedy algorithm but with less time complexity.

1. Introduction

Social networks play an important role in information diffusion, products adoption, as well as public opinion monitoring, etc. As an interesting research topic, influence maximization aims to select a subset of k most influential nodes as seed set to maximize the influence propagation. Identifying the top-k influential nodes effectively and efficiently to maximize the

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

https://doi.org/10.1016/j.physa.2018.09.040
0378-4371/

expected marginal return is of great significance not only for theoretical research of social network analysis, but also for very promising practical applications, such as viral marketing [1,2], critical infrastructure monitoring [3] and outbreak prediction of cyberspace security [4], etc. Research on influence maximization has received tremendous attention of multidisciplinary researchers and engineers in recent years, and many significant works have been reported.

Originating from viral marketing, which takes advantage of the "word-of-mouth" effect in the promotion of innovation products [1], influence maximization has been proved to be a NP-hard problem [5]. Thus, how to identify the top-k influential nodes effectively remains an important and challenging problem. Submodularity-based methods including greedy algorithm can guarantee the identification quality, but the time complexity is unbearable especially in large-scale networks. Meanwhile, conventional centrality-based measures always achieve satisfying results when identifying single influential node, but they tend to return suboptimal solutions to multiple influential nodes identification. In this paper, we focus on studying the sensitivity of network topology-based local search strategy for the metaheuristic discrete particle swarm optimization (DPSO) adopted to solve influence maximization, and find the original DPSO is troubled by premature easily, in other words, it always fails to select a global seed set of influential nodes. Therefore, an enhanced discrete particle swarm optimization is proposed in this paper for the identification of top-k most influential nodes for influence maximization. Our contributions are summarized as follows:

- We make use of local network topology characteristics of seed nodes and present an enhanced topology-based local search strategy to improve the local exploitation ability of DPSO.
- The enhanced local search strategy is integrated into DPSO framework to identify the top-k global influential nodes more effectively for influence maximization in social networks.
- The experimental results on six different scale of social networks under independent cascade model show that ELDPSO achieves comparable results to greedy algorithm and can provide stable solution of identification with a low time complexity.

The rest of this paper is organized as follows: problem description and related studies on influence maximization are given in Section 2. Section 3 presents the enhanced topology-based local search strategy for discrete particle swarm optimization. Section 4 gives the experimental results and analysis. Conclusions are drawn in Section 5.

2. Related works

Domingos and Richardson [6] studied the expected profit from the customers of viral marketing firstly in a network perspective and modeled influence maximization as a Markov random field. According to the proposed mathematical model, influence maximization can be generally defined as follows.

Definition 1. Let $G = (V, E)$ be a graph with $|V| = n$ nodes and $|E| = m$ edges. Given an integer $k < |V|$, the target of influence maximization is to select k most influential nodes as the initial seed set S so as to the expected number of activated nodes, denoted as influence spread $\sigma(S)$, is maximum under a cascading propagation model.

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

where S^* is the best seed set returned by the critical function.

Kempe et al. [5] formulated influence maximization as a combinatorial 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 guarantee $(1 - 1/e - \epsilon)$ approximation of the optimal solution was proposed. However, the greedy algorithm has to call an estimator function to calculate the new expected influence spread $\sigma(S \cup \{u\})$ and the marginal return when a candidate node u is added to seed set S in each step and this is a #P-hard problem [7,8]. Besides the calculation, tens of thousands Monte-Carlo simulations have to be simulated in each step to get an acceptable approximate solution, which is time-consuming and seriously limits its application in large-scale networks in practice. Substantial works based on this effort have been done, and many state-of-the-art algorithms were presented to reduce the time complexity [9–12]. CELF, proposed by Leskovec et al. [3], is one of the typical submodularity-based algorithms. The authors exploited the property of *submodularity* to reduce the number of influence evaluation with lazy update of the marginal return for each node, which is about 700 times faster than the simple greedy algorithm. However, the experimental results showed that CELF is still time-consuming especially in random networks because the marginal return for each node in the priority queue is approximately Gaussian distribution and tens of thousands Monte-Carlo simulations have to be conducted for quite a fraction of the node set. Kimura et al. [9] stated that the set of examined nodes are the same as the set of active nodes in the process for each simulation, and provided an efficient greedy algorithm to estimate the expected marginal influence $\sigma(S \cup \{v\})$ on the basis of bond percolation and graph theory with a large reduction in computational cost.

Chen et al. [13] argued that when we select a node u as a seed, the degree of its neighbor node v should be discounted to a certain amount. Two heuristics named SingleDiscount (SD) and DegreeDiscount were proposed to select seed nodes in a more efficient way. Meanwhile, the authors indicated that inefficiency is the intrinsic drawback of greedy-based algorithms, and pointed out that developing efficient seed set selecting schemes based on network topology may be a promising way. Shang et al. [14] consented that community-based influence maximization algorithm are generally faster than traditional greedy algorithms. To reduce the time consumption, they derived a constant sum evaluation form of the total influence spread and proposed a community-based framework for influence maximization (CoFIM) under weighted cascade (WC) model scalable to large-scale networks. The simulations showed that community-based algorithm generally performs better than traditional greedy-based algorithms based on WC model. To reduce the calculation for evaluating influence spread in large-scale networks, Kim et al. [15] proposed a pruning and parallelizable method for influence maximization based on *Random Walk* and *Rank Merge* schemes. The results showed that the method can dramatically reduce the consuming time. In addition, Kim et al. [16] proposed an algorithm to process update operations of dynamic graphs based on reachability sketches. The algorithm can achieve comparable results to conventional heuristics. Huang et al. [17] modified the *SSA* [18] algorithm based on sampling method and proposed an effective named *SSA-Fix* algorithm for influence maximization more recently.

Conventional centrality measurements [19,20] including high degree centrality (DC) [5,21], betweenness centrality (BC) [22], closeness centrality (CC) [23] and *k-shell* decomposition (K-shell) [24], etc., can select a seed set within a satisfying short time. Different with greedy-based algorithms that select seed nodes one by one, the top-k influential nodes are selected simultaneously and consistently by these centrality-based methods. DC, a general and simple local measure, can perform well in minimum influential nodes identification. BC and CC are two global measures that identify influential nodes according to the whole network topology so that they are not scalable to large-scale networks due to the large computation. Kitsak et al. [25] proposed the *K-shell* decomposition method and pointed out the distance between spreaders becomes the crucial parameter that determines the extent of influence spreading. However, the *K-shell* method tends to assign many nodes that have different propagation capability to the same *KS* value, which is the primary cause of overlapped influence propagation. Thus, Garas [26] reiterates that less efficiency is a particular bugbear accompanying the *K-shell* decomposition method when identifying multiple seed nodes simultaneously. Recently, Bian et al. [27] proposed a node information dimension (NID) to identify influential nodes by calculating each segmental local dimension at different distance for each node. The NID measurement is more accurate than the CC method, but its computational complexity is the same as the BC and CC, which makes it inappropriate to be applied in large-scale networks.

Metaheuristic algorithms, which mimic the behaviors of biotic population or characters of physical phenomena, have been adopted widely in solving combinational optimization problems [28–30] for the intrinsic effectiveness, robustness and simplicity. In the last few years, researchers have been trying to apply metaheuristic algorithms to solve influence maximization problem in social networks. As far as we know, Jiang et al. [31] adopted simulated annealing (SA) algorithm firstly for influence maximization. The experimental results showed that SA runs faster than greedy algorithm by 2∼3 orders of magnitude while being able to improve the accuracy of greedy algorithm. Gong et al. [32] considered that spreader's influence is time-decaying so that the influence spread of each spreader is limited within a local area. Based on the ideology that a node's influence within two-hop area contributes a large proportion of its global influence, they constructed a local influence estimator (*LIE*) optimization function model to evaluate the propagation ability of each node in the network, and presented a discrete particle swarm optimization (DPSO) to optimize the *LIE* fitness function. Sankar et al. [33] proposed a swarm intelligence algorithm based on the study of bees' *waggle dance* behavior for influence maximization, and stated that understanding the swarm intelligence in biology society may be a practical way to design efficiency influence maximization algorithms. As emphasized in Ref. [34,35], metaheuristic algorithms always suffer from local optimal solution. In other words, local search strategy plays a crucial role in helping metaheuristic algorithms converge to global optimization.

In this paper, we employ the bio-inspired particle swarm optimization for influence maximization. To avoid DPSO from being trapped into local optimal seed set easily, the sensitivity of local search strategy for DPSO is studied prudently and an enhanced network topology-based local search strategy is designed to enhance the exploitation capability of DPSO when identifying the top-k influential nodes.

3. Method

Particle swarm optimization (PSO), a bio-inspired metaheuristic algorithm mimicking the foraging behavior of bird flocks, was proposed by Kennedy and Eberhart [36]. PSO has been widely applied to tackle with optimization problem [37–40] for its effectiveness and robustness. The original mathematical model of PSO can be formulated as follows:

$$V_i^{t+1} = V_i^t + c_1 r_1 (Pbest_i - X_i^t) + c_2 r_2 (Gbest - X_i^t) \tag{2}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{3}$$

where $V_i = (v_{i1}, v_{i2}, \ldots, v_{id})$ and $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ represent the position and velocity vectors of particle i ($i = 1, 2, \ldots, N$) at time t respectively, where N is the swarm size. N and d are the swarm size and the dimension of problem space respectively. c_1 and c_2 are two constant learning factors. r_1 and r_2 are two random numbers drawn uniformly from $(0, 1)$. $Pbest_i$ denotes the previous historical best position of particle i, and $Gbest$ denotes the global best position of the swarm up to current generation.

Among the state-of-the-art works on PSO, Shi et al. [38]. proposed a notable weighted PSO (WPSO) and gave the specification of parameters evaluation to assure the WPSO can converge to global optimal solution. Following this seminal literature, a variety of local search strategies to improve the convergence of WPSO were proposed. Recently, Zhang et al. [41] proposed a competitive mechanism to optimize the global evolution. Zhao et al. [42] proposed a decline disturbance index strategy to enhance the performance of PSO and applied the enhanced PSO to solve multi-objective job-shop scheduling problem. For more details of PSO, please see Appendix A.

As mentioned in Section 2, Gong et al. [32] proposed a DPSO to solve influence maximization problem for the first time. Based on Two-Degree Theory [43], a local influence estimator (*LIE*) function shown in Eq. (4) was modeled to estimate the influence spread within the two-hop area of a node.

$$\begin{aligned} LIE &= \sigma_0(S) + \sigma_1^*(S) + \tilde{\sigma}_2(S) \\ &= k + \sigma_1^*(S) + \frac{\sigma_1^*(S)}{|N_S^{(1)} \backslash S|} \sum_{u \in N_S^{(2)} \backslash S} p_u^* d_u^* \\ &= k + \left(1 + \frac{1}{|N_S^{(1)} \backslash S|} \sum_{u \in N_S^{(2)} \backslash S} p_u^* d_u^*\right) \sum_{i \in N_S^{(1)} \backslash S} \left(1 - \prod_{(i,j) \in E,\, j \in S} (1 - p_{i,j})\right) \end{aligned} \tag{4}$$

where $\sigma_0(S)$ is the k most influential nodes in the seed set S, $\sigma_1^*(S)$ and $\tilde{\sigma}_2(S)$ are the expected influence spread of one-hop and two-hop area of the seed set respectively. Correspondingly, $N_S^{(1)}$ and $N_S^{(2)}$ represent the one-hop and two-hop area of seed set S. p_u^* is a small constant active probability of different diffusion models. d_u^* is the number of edges of node u within $N_S^{(1)}$ and $N_S^{(2)}$. Apparently, the influential node seed set selection was converted into an optimization problem targeting to find a seed set of k influential nodes to maximize the *LIE* function value, and the DPSO was proposed to optimize the *LIE* function. The updating rules of velocity and position vectors are designed in a discrete form as follows.

$$V_i \leftarrow H(\omega V_i + c_1 r_1 (Pbest_i \cap X_i) + c_2 r_2 (Gbest \cap X_i)) \tag{5}$$

$$X_i \leftarrow X_i \oplus V_i \tag{6}$$

where X_i is coded by the integer IDs of k candidate nodes, and so do the $Pbest_i$ and $Gbest$. Operator "∩" is defined as a logical similar intersection operation, and the result returned by the operator is a vector composed by 0 and 1, where 1 represents the element in the corresponding position of X_i has an identical element in $Pbest_i$ or in $Gbest$, 0 is the otherwise. Velocity vector V_i is encoded by 0 or 1, where 0 represents the node corresponding to the position of X_i is an influential node, 1 represents the node corresponding to the position of X_i needs to be replaced or to be adjusted. $H(\cdot)$ is a decision function to ensure that $v_{ij} \in V_i$ is 0 or 1, where $j = 1, 2, \ldots, k$. For more details of the operations for DPSO, please see Appendix B.

3.1. An enhanced local search strategy

To avoid the DPSO algorithm exploring blindly, The authors [32] suggested a network-specific local search strategy given in Algorithm 1 to guide particles converge to global optimization. Where function $Replace(\cdot)$ is used to replace x_{bi} with one node extracting randomly from its direct neighbor set.

Algorithm 1 Pseudocode of Local search

Input: Particle X_a
$X_b \leftarrow X_a$
for each element $x_{bi} \in X_b$ **do**
  $Flag \leftarrow FALSE$
  $Neighbors \leftarrow N_{bi}^{(1)}$
  repeat
    $x_{bi} \leftarrow Replace(x_{bi,Neighbors})$
    if $LIE(X_b) > LIE(X_a)$ **then**
      $X_a \leftarrow X_b$
    else
      $Flag \leftarrow TRUE$
    end if
  until $Flag$ is $TRUE$
end for
Output: Particle X_b

According to the local search strategy given in Algorithm 1, we can see that the scheme is designed as a semi-greedy strategy to help the algorithm converges rapidly. As expected, there are two scenarios about the terminal condition of the *repeat* operation. The first one is that if the replacement operation is successful, i.e., $LIE(X_b) > LIE(X_a)$, then the *Replace* function will be executed iteratively until the value of *Flag* is *TRUE*, then the strategy may return a normal solution. The second one, however, may be an unsatisfying solution. In this scenario, if it fails to replace x_{bi} at the first time, that is, $LIE(X_b) < LIE(X_a)$, then *TURE* will be assigned to parameter *Flag*. The local search process will be terminated immediately and contributes nothing to the current generation. What follows is that the algorithm tends to miss a more influential candidate node and to be trapped into a suboptimal seed set. Therefore, improving the greedy-based local search strategy can help DPSO find potential promising influential nodes. In addition, the replacement way of x_{bi} by its direct neighbor set is done with no restrictive factors, in other words, the node in the neighbor set is taken out in an unordered way to replace x_{bi}, which may also result in unsatisfying replacement.

How to balance the exploration and exploitation capabilities remains a primary task on designing an effective optimization algorithm. Given a network structure, it is feasible to replace a candidate node with a neighbor node that has higher propagation capability. Therefore, taking full advantage of the topology structure, especially the local interaction relationships among nodes, to get the best seed set is a promising scheme for designing an effective local search strategy.

Take the Karate club network shown in Fig. 1(a) for an example, we encode the 34 nodes in the network with a positive integer and let $S = \{5, 12\}$ be the *Gbest* of current generation. If we apply the original local search strategy on the seed set, the one-hop neighbors of node 5 will be obtained firstly, denoted as $N_5^{(1)} = \{1, 7, 11\}$. Then node 5 will be replaced by node 1 firstly because $LIE(\{1, 12\}) > LIE(\{5, 12\})$, that is, node 1 can bring more marginal return than that of node 5, and the replacement will be terminated according to the original local search scheme leaving subset $\{11\}$ without being considered. Secondly, node 12 will be processed sequentially. Node 12 has only one direct neighbor node 1, which is the direct neighbor of node 5, and has already been selected as a seed, finally, the returned optimal solution is a novel seed set $S' = \{1, 12\}$, and the corresponding *LIE* value according to Eq. (4) is approximately $LIE = 2 + (1 + 0.85/15) * 0.75 \approx 2.79$ under IC model with a propagation probability $p = 0.05$. However, if we recombine the temporary optimal seed set by node's degree firstly, i.e., the ordered seed set is $S = \{12, 5\}$. Then a local greedy strategy, as shown in Algorithm 2 is employed to replace the current candidate node with the most influential node from the direct neighbor set of each node from temporary seed set, and the result will be radically different. Firstly, we recombine the temporary seed set by node's degree in an ascending order and we have a ordered seed set $S = \{12, 5\}$. Secondly, the one-hop neighbors of node 12 will be obtained preferentially, denoted as $N_{12}^{(1)} = \{1\}$, there is only one neighbor, and node 12 will be replaced by node 1 according to the value of $LIE(\{1, 5\}) > LIE(\{12, 5\})$. Thirdly, the direct neighbors of node 5 will be obtained, denoted as $N_5^{(1)} = \{1, 7, 11\}$. For node 1 has been selected as a seed node, the marginal return of replacing node 5 with node 7 is more than that brought by node 11, so the returned seed set is $S' = \{1, 7\}$ and the corresponding *LIE* value is approximately $LIE = 2 + (1 + 0.75/16) * 0.895 \approx 2.94$.

The two similar seed sets S' result in different *LIE* value though they have the same two-hop areas in these two scenarios. The key distinction is that node 7 has more out-degrees and contributes more influence propagation than node 12 in one-hop area before the node's influence decays with farther pairwise paths. That is, the original DPSO tends to fail to exploit a more promising seed set when we apply the original local search algorithm on the supposed seed set $S = \{5, 12\}$ directly. Furthermore, the local search strategy probably leaves a fraction of top-k influential nodes out of the global best seed set when the seed size k is large when dealing with large-scale social networks.

From the illustration shown in Fig. 1, we can see that different replacement strategies result in diverse *LIE* value, and the *LIE* value may be extremely different when the seed size k is large. Therefore, it is of great important to design a robust local search strategy prudently so that the special local relationships among the candidate seed nodes can be further exploited.

Inspired by the above illustration, we try to take full use of network topology and propose an enhanced network topology-based local search strategy to improve the local exploitation capability of DPSO. Firstly, the k nodes in the temporary optimal seed set of each iteration are recombined in ascending order by degree metric to let the nodes with lower degree centrality exploit more influential neighbors preferentially. Secondly, a local greedy strategy is presented to replace the current candidate node with the most influential node from the direct neighbor set of each node in the temporary seed set. The pseudocode of the enhanced local search strategy is given in Algorithm 2. Where, function $degree(Gbest^*)$ returns the degree centrality of each node in the current global best particle $Gbest^*$. Function $Order(Gbest^*, d_{Gbest^*})$ returns an ordered node set on the basis of the degree centrality of each node in ascending order. Function $Length(Nei\_set)$ returns the size of the neighbor set Nei_set. $Replace(gbest_i, Nei\_set)$ is adopted to replace $gbest_i$ with one of its one-hop area neighbors and guarantees that there is no repeated nodes after the replacement is finished.

3.2. ELDPSO for influence maximization

According to the framework of DPSO, the updating rules of position are refereed as the exploration part to search optimal solution globally in the problem space. A local search operator acts itself as the exploitation part to improve the solution quality locally. We integrate the enhanced local search strategy into the basic DPSO framework and present an enhanced discrete particle swarm optimization (we call it ELDPSO for short) for influence maximization in social networks. That is, once the temporary optimal seed set of current iterative generation is obtained, we apply the enhanced local search strategy on the seed set subsequently to exploit a better optimal seed set for the evolution of next generation until the termination condition is satisfied. The framework of ELDPSO is modeled as in Algorithm 3.

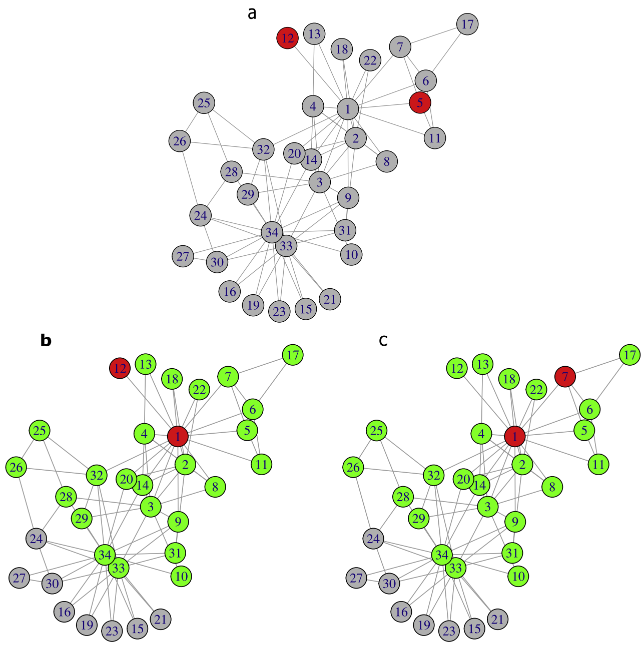


Fig. 1. Illustration of two different local search strategies for calculating *LIE* on Karate network. (a) Given a temporary optimal seed set $S = \{5, 12\}$ of one generation, red nodes represent the active seed set S, gray nodes represent inactive nodes, and green nodes represent the two-hop area neighbor set. (b) If we apply the original local search strategy on $S = \{5, 12\}$, $S' = \{1, 12\}$ will be returned as the new seed set after the termination of local search and $LIE(\{1, 12\}) \approx 2.79$ with a propagation probability $p = 0.05$. (c) If we apply the latter local search strategy on the initial S, the returned seed set is $S' = \{1, 7\}$ and $LIE(\{1, 7\}) \approx 2.94$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Algorithm 2 An Enhanced local search strategy($Gbest^*$)

Input: $Gbest^*$ of current iteration.

$d_{Gbest^*} \leftarrow Degree(Gbest^*)$
$Gbest' \leftarrow Order(Gbest^*,\ d_{Gbest^*})$
for each element $gbest_i \in Gbest'$ **do**
  $Nei\_set \leftarrow N^{(1)}_{gbest_i}$
  $index \leftarrow 1$
  $len \leftarrow Length(Nei\_set)$
  while $index \leq len$ **do**
    $Gbest' \leftarrow Replace(gbest_i,\ Nei\_set)$
    if $LIE(Gbest') > LIE(Gbest^*)$ **then**
      $Gbest^* \leftarrow Gbest'$
    end if
    $index \leftarrow index + 1$
  end while
$Gbest' \leftarrow Gbest^*$
end for
return the newly global best seed set $Gbest'$

3.3. Computational complexity of ELDPSO

Computational complexity is one of the main metrics for evaluating the performance of algorithms for influence maximization problem. According to the framework of ELDPSO, The computational complexity of local search strategy is one of the major factors affecting the efficiency of ELDPSO. For the enhanced local search strategy, the running time lies in the ranking operation and the greedy-based replacement scheme. The ranking operation requires $O(k \cdot \log k)$, and the greedy-based replacement scheme requires $O(k \cdot \bar{D})$, where $\bar{D}$ is the average degree centrality of a network, thus, the whole time complexity of the redesigned local search is $O(k(\log k + \bar{D}))$. In addition, the degree-based initialization needs $O(N \cdot k)$ operations. The updating of velocity needs $O(k \cdot \log k \cdot N)$ operations. The updating of position X and $Pbest$ need $O(k \cdot N)$

Algorithm 3 Framework of ELDPSO for Influence Maximization

Input: Graph $G = (V, E)$, maximum number of iterations g_{max}, particle swarm size N, the inertia weight ω, learning factors c_1 and c_2.

Initialize iterator $g = 0$
Initialize position vector X based on degree centrality heuristic method with random disturbance
Initialize *Pbest* vector based on degree centrality heuristic method with random disturbance
Initialize velocity vector $V \leftarrow 0$
Select out the initial global best position vector $Gbest^*$ according to the *LIE* value of each X_i
while $g < g_{max}$ **do**
  Update the velocity vector V according to Eq. (5)
  Update the position vector X according to Eq. (6)
  Update the *Pbest* and select out the best $Gbest^*$ of the current generation
  Employ the enhanced local search operation: $Gbest' \leftarrow eLS(Gbest^*)$
  Compare and update the $Gbest^*$: $Gbest^* \leftarrow \max(Gbest^*, Gbest')$
  $g \leftarrow g + 1$
end while
return $Gbest^*$ as the seed set S

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

Networks	$\|V\|$	$\|E\|$	$\langle k \rangle$	$\bar{d}$	C	AC
Karate [44]	34	78	4.588	2.408	0.588	−0.476
NetScience [45]	379	914	4.823	6.042	0.798	−0.082
Email [46]	1 133	5 451	9.622	3.606	0.254	0.078
Blogs [47]	3 982	6 803	3.417	6.252	0.493	−0.133
CA-HepTh [48]	9 877	25 998	5.264	5.945	0.600	0.268
PGP [47]	10 680	24 316	4.554	7.486	0.440	0.238

operations. Evaluating *LIE* needs $O(k \cdot \bar{D})$ operations. The others need one unit operation separately. Thus, the upper bound computational complexity of ELDPSO is $O(k^2 \cdot \log k \cdot N \cdot \bar{D}^2 \cdot g_{max})$.

4. Experiments and analysis

In this section, we conduct extensive experiments in six real-world social networks shown in Table 1 to verify the performance of the proposed ELDPSO. All simulation procedures are coded by C++ language and executed on a PC platform with 4 Intel (R) Cores (TM) , i7-4790 3.60 GHz CPU and 8G memory.

4.1. Baseline algorithms

The experimental work is divided into two separate phases. Firstly, comparison on *LIE* are carried out on the six social networks to verify the effectiveness difference between ELDPSO and original DPSO algorithm. Secondly, we compare the performance of ELDPSO with seven other baseline algorithms including greedy algorithm, CELF, DPSO, SD, DD, DC and BC, which are briefly described as below, on identifying the top-k most influential nodes for influence maximization.

• The greedy algorithm [5], which has been introduced in detail in Section 2, selects the seed node that can bring the maximum marginal gain in each of the k iterations.

• The Cost-Effective Lazy Forward (CELF) [3] is a greedy algorithm with a "lazy forward" optimization strategy by exploiting the submodularity property.

• The original discrete particle swarm optimization (DPSO) is a metaheuristic algorithm that selects seed nodes via the cooperative evolution of the particle swarm based on a given influence spread estimation function model.

• The SingleDiscount (SD) heuristic [13] is a modified degree centrality-based algorithm. In each step, the algorithm selects the node with the highest degree centrality as a seed node, and discounts the degree of its neighbors those have not been selected as seed node to a amount.

• The diffusion degree (DD) heuristic, which was proposed by Kundu et al. [21], is a semi-local centrality-based measure relying on IC model. DD considers the expected contribution of a node v and the cumulative contribution by the neighbors of node v in the diffusion process with propagation probability λ_v, and selects the top k nodes with the highest DD value as a seed set according to Eq. (7).

$$C_{DD}(v) = \lambda_v * C_D(v) + \sum_{i \in neighbors(v)} \lambda_i * C_D(i) \tag{7}$$

• The degree centrality (DC) is a local centrality measurement which selects top-k nodes with the highest degree centrality as seed nodes.

• The betweenness centrality (BC) is a global centrality measurement which selects top-k nodes with the highest betweenness centrality as seed nodes.

We study and implement the experimental simulations based on the IC model in two probability scenarios $p = 0.01$ and $p = 0.05$ in separate trails. For the original greedy algorithm and CELF, we run the Monte-Carlo simulation 10,000 times, and the simulation times for other six algorithms are set to 1000 respectively to obtain the average influence spread. For Karate network, the swarm size N and the maximum seed size k are both set to 20 separately for ELDPSO and DPSO due to it has only 34 nodes. For other social networks, N and maximum k are set to 30 accordingly. The generation number g_{max} for ELDPSO and DPSO is set to 100. The parameter λ used in the DD method is the same as the propagation probability p, and it is set to 0.01 and 0.05 in separate trails.

4.2. Diffusion model

We study and conduct the experimental simulations based on the IC model intensively. IC model is a classic probability model mimicking information diffusion in social networks formalized by Kempe et al. [5]. According to IC model, for an active node u at step t, it has only one chance to activate each of its direct inactive neighbor v and succeed with a probability p_{uv}. 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 successfully, then v will be active and has one chance to activate its direct inactive neighbors from step $t + 1$. The diffusion process will stop if no node is activated at step T.

4.3. Comparison on LIE evaluating

To verify the effectiveness of the enhanced local search strategy, we conduct experiments to compare ELDPSO with original DPSO. As evidenced by the simulations in Ref. [32], we take on the suggestion and set the learning factors c_1 and c_2 to 2, the inertia weight $\omega = 0.8$, respectively.

In addition, we also consider the effect of two other global centrality measures including BC and CC on the improvement of local search strategy while recombining the temporary seed nodes according to the two global centralities separately. Therefore, two other algorithms named ELDPSO-BC and ELDPSO-CC are presented and experiments on evaluating *LIE* value are carried out. Paraphrasing, the enhanced local search strategy in ELDPSO-BC recombines the potential seed nodes in the current optimal seed set of each generation by the betweenness centrality metric of each node in ascending order, i.e., the nodes with lower betweenness centrality have prior chances to exploit more influential nodes. A similar operation is adopted by ELDPSO-CC, in which the potential seed nodes in the current optimal seed set of each generation are recombined by the closeness centrality of each node in ascending order, i.e., the nodes with lower closeness centrality have prior chances to exploit more influential nodes. To calculate nodes' BC and CC, a fast method proposed by Brandes [49] is adopted, of which the time complexity is $O(nm)$, where n is the number of nodes and m is the number of edges in a network.

As the evolutionary processes shown in Fig. 2, we can see firstly that, when the seed set size $k < 10$, all the four metaheuristic algorithms are effective to select k most influential nodes for they achieve almost the same *LIE* value under different k scenarios in each subfigure. However, compared to DPSO, all the other three enhanced algorithms based on DC, BC and CC perform more satisfying *LIE* value under both of the two propagation probabilities $p = 0.01$ and $p = 0.05$ in the six social networks when $k > 10$. Secondly, both ELDPSO-BC and ELDPSO-CC can achieve comparable *LIE* value to ELDPSO in all the scenarios in Fig. 2. However, ELDPSO performs apparently as the best one in optimizing *LIE* estimation function with approximately 15.9% higher *LIE* value than that achieved by DPSO when $k = 30$ in the PGP network shown in Fig. 2(l). It means that recombining the nodes in the temporary optimal seed set of each generation in ascending order by node degree centrality contributes more improvement on DPSO than that by betweenness centrality and closeness centrality. That is because the number of edges in a local two-hop area plays an important role in spreading a node's influence according to the Two-Degree Theory [43] . Thirdly, compared to other three algorithms, we can see from Fig. 2 that DPSO always provides unstable *LIE* solutions, especially in Fig. 2(k) and (l). As to Fig. 2(k), the evolutionary curve simulated by DPSO is choppy, which implies the original DPSO is unstable in a large-scale network. Besides the drawback, Fig. 2(l) shows that DPSO tends to be trapped into local optimal seed set easily when the seed size $k \geq 15$. All the observable phenomena demonstrate that an effective local search strategy plays an essential role in the approach of selecting global best seed set when employing metaheuristic algorithms to solve influence maximization.

Though ELDPSO-BC and ELDPSO-CC achieve comparable *LIE* value to ELDPSO, the time complexity of the former two algorithms is higher than that of ELDPSO. The degree centrality of each node can be obtained in constructing the adjacent matrix of a network. However, it needs at least extra $O(nm)$ to calculate the betweenness centrality and closeness centrality of each node in the network, and the time complexity of ELDPSO-BC and ELDPSO-CC is $O(nm + k^2 \cdot \log k \cdot N \cdot \bar{D}^2 \cdot g_{max})$ respectively.

Fig. 3 shows the comparison on running time of the four metaheuristic algorithms when to select $k = 30$ (for Karate, $k = 20$) seed nodes under propagation probability $p = 0.05$ in the six social networks, where we only consider the time once the algorithm converges to global optimal *LIE* value. From Fig. 3 we can see that the original DPSO needs the least time in the six networks compared to other three improved algorithms, but it tends to be trapped into local optimal easily as

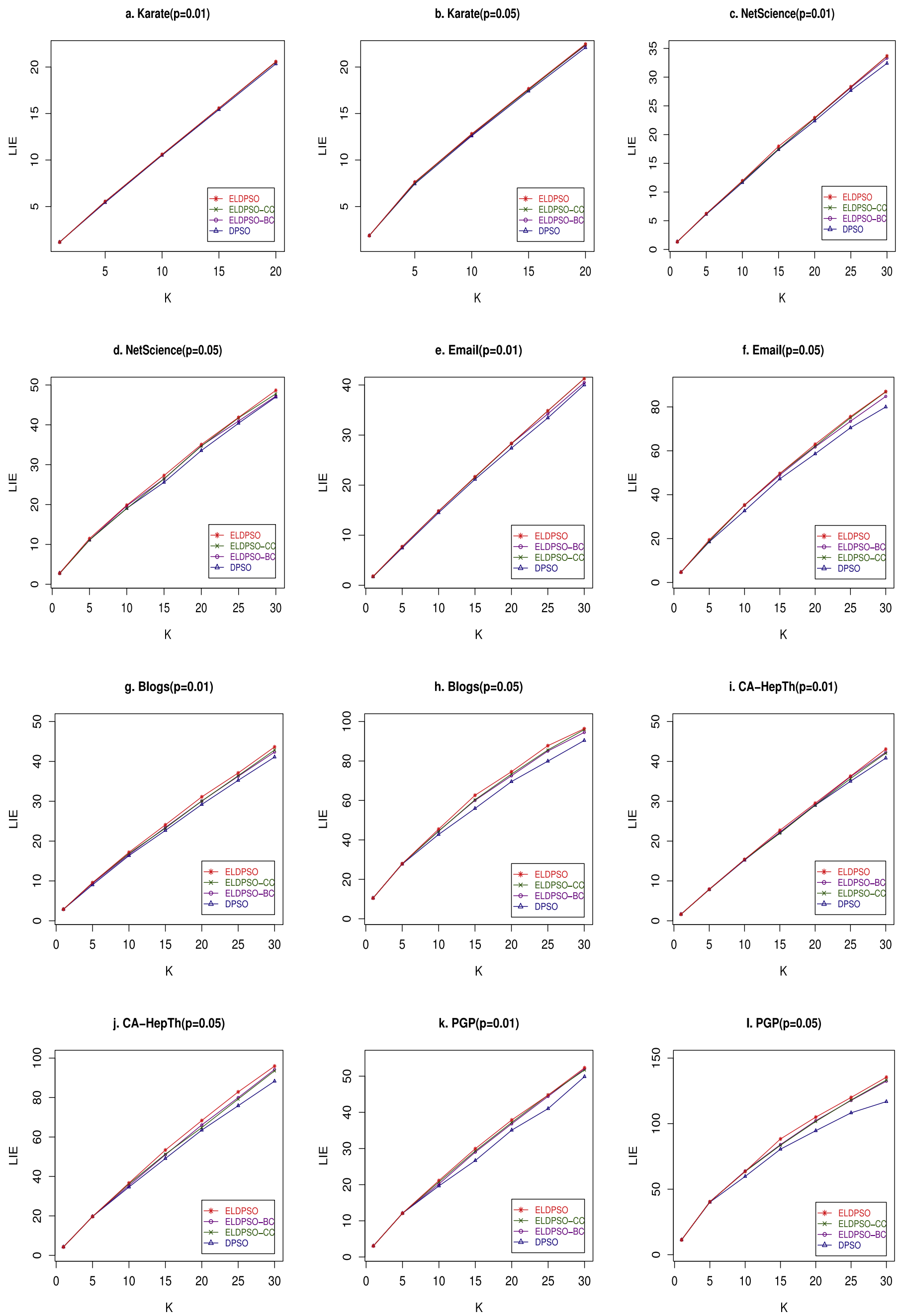


Fig. 2. Comparison on *LIE* value of different seed size k simulated by the four metaheuristic algorithms under propagation probabilities $p = 0.01$ and $p = 0.05$ in the six social networks.

shown in Fig. 2 for the deficient local search strategy. Meanwhile, both ELDPSO-BC and ELDPSO-CC are inefficient because they need much more time to calculate the betweenness centrality and the closeness centrality of each node in the network, which restricts the scalability of the two algorithms to large-scale networks. Therefore, we are inclined to choose ELDPSO, which can well balance the effectiveness and efficiency, to select the k most influential nodes in the six social networks and compare its effectiveness with seven other baseline algorithms on influence spread under two propagation probabilities $p = 0.01$ and $p = 0.05$.

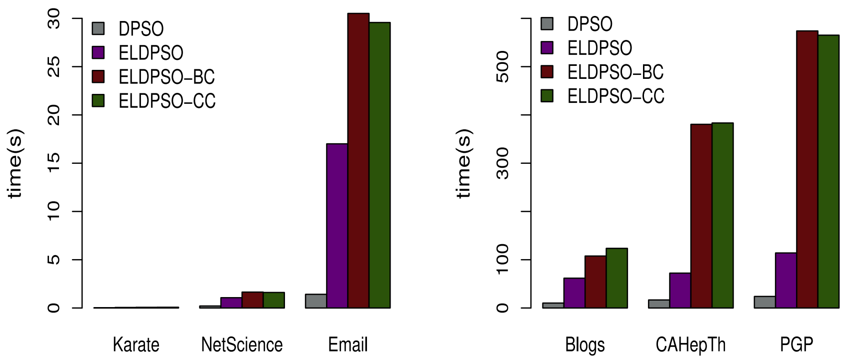


Fig. 3. Comparison on running time of the four metaheuristic algorithms when to select $k = 30$ (for Karate, $k = 20$) seed nodes under propagation probability $p = 0.05$ in the six social networks.

4.4. Comparison of typical algorithms

To verify the effectiveness of ELDPSO, we choose seven other algorithms mentioned in Section 4.1 as baselines. The simulated evolutionary results shown in Fig. 4 in six real social networks under IC model illustrate the influence spread of the eight algorithms.

From the simulated curves, we can see firstly that ELDPSO achieves comparable results to that obtained by greedy algorithm and CELF. Remarkably, ELDPSO performs better than greedy algorithm in Fig. 4(a), (c) and (e). In the scenarios of Fig. 4(c),(h) and (k), ELDPSO achieves even better results than CELF, which proves ELDPSO is effective for influence maximization when the propagation probability is small. In addition, ELDPSO outperforms apparently as a more satisfying and robust algorithm than the original DPSO and other four centrality-based heuristics in the six social networks. Obviously, the effectiveness and improvement of ELDPSO benefits from the enhanced local search strategy, through which more influential nodes can be exploited from the local topology structure of the temporary seed nodes of each generation.

As shown in Fig. 4, the gaps illustrated by the influence spread curves of each network become wider between the given two propagation probabilities. When p=0.01, all the eight algorithms spread almost the same size on each of the six social networks when $k < 5$, separately. However, the curves are variant with the increase of k. When p=0.05, the difference among the eight algorithms is more apparent even since from $k = 1$ such as in Fig. 4(f), (h), (j) and (l). Among the eight algorithms, greedy algorithm performs itself almost as the best one in the six networks due to its hill-climbing strategy. Following up greedy algorithm, ELDPSO achieves comparable results to greedy algorithm and better results than DPSO. As discussed in Section 3, the enhanced local search strategy plays a significant role in helping the algorithm exploit more promising seed nodes.

In terms of the seed size k, the eight algorithms achieve almost the same influence spread size when $k < 5$. With the increase of k, submodularity-based and metaheuristic algorithms can perform satisfying solution effectively and robustly all the time. As expected, conventional topological-based centralities DC and BC perform well in individual influential node identification, but they cannot provide guarantee solution especially when the seed size k is large. More often than not the reason might be found in the principles on which the centrality-based methods rely. They tend to select the k nodes with such as highest degree centrality value, or the largest betweenness value, etc., based on network topology. And the active nodes influenced by the nodes from seed set returned by these methods tend to be overlapped, which will result in a limited influence spread size.

It is worth to mention that SD algorithm performs well in density networks, but it tends to achieve suboptimal solutions in sparse networks, such as shown in Fig. 4(l). The DD algorithm that considers the expected contribution of a given node and the cumulative contribution by the neighbors of the node seems to be a reasonable influence estimation metric, but it always assign a same diffusion degree value the nodes that have different degree centrality. Therefore, the seed set selected by DD is always a suboptimal one, especially in Fig. 4(j) and (l), where the influence spread size increases inappreciably with the seed size $k > 15$. Besides the comparisons on the average influence propagation of the eight algorithms, we also give further exploration using box plots in Appendix C.

4.5. Performance metric

In order to evaluate the performance of ELDPSO and other algorithms statistically, we adopt a Precision@k [50] metric, of which the ideology is expressed as follows, to measure the proportion of the top-k influential seed nodes identified by these algorithms to the best seed set approximately. Given two seed sets S^1 and S^2 representing the top-k influential nodes selected by two algorithms respectively. Precision@k is defined as the ratio of the number of common seed nodes of the two sets over set size k. We select the algorithm that performs the best among the eight algorithms when identifying top $k = 30$

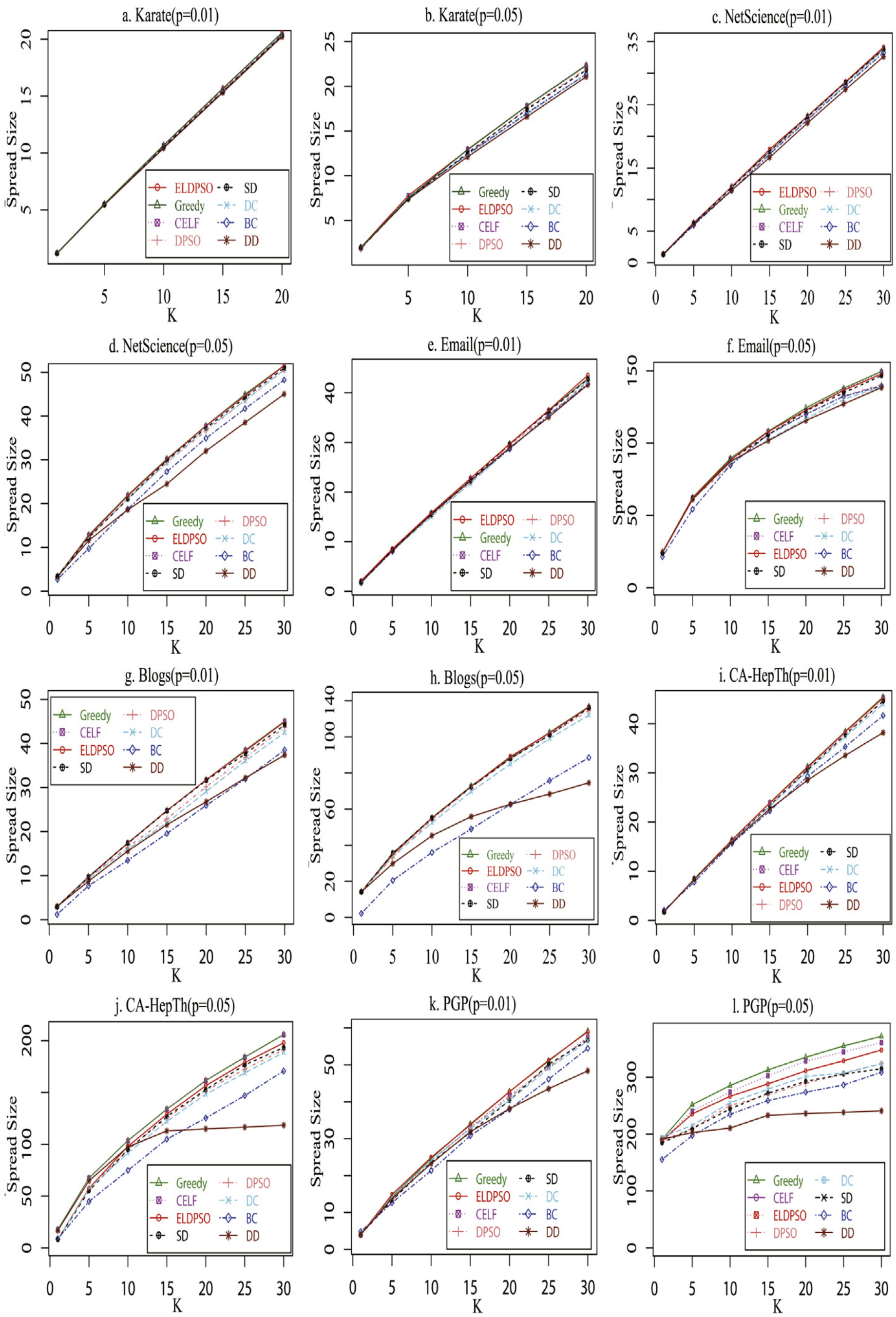


Fig. 4. Influence spread size of different seed size k simulated by eight algorithms under IC model with two propagation probabilities $p = 0.01$ and $p = 0.05$ in the six social networks.

(for Karate network, $k = 20$) influential nodes under each probability scenario in the six networks as the baseline algorithm, denoted by symbol '-'. Table 2 shows the statistical results based on the Precision@k metric.

$$precision@k = \frac{|\{V_i | V_i \in S^1, V_i \in S^2\}|}{k} \tag{8}$$

As illustrated in Section 4.4, greedy algorithm performs almost as the best one compared to other seven algorithms, so we treat the seed set selected by greedy algorithm as the global optimization except three scenarios shown in Fig. 4(a), (c) and (e), where ELDPSO is considered as the best one.

Table 2
Statistic result of the eight algorithms based on Precision@k metric in the six social networks.

Networks	p	Greedy	ELDPSO	CELF	DPSO	SD	DC	BC	DD
Karate	0.01	**0.95**	–	0.650	0.700	0.750	0.650	0.600	0.600
	0.05	–	**0.800**	0.600	0.750	0.550	0.600	0.550	0.500
NetScience	0.01	**0.900**	–	0.867	0.800	0.800	0.700	0.650	0.433
	0.05	–	0.667	0.600	0.533	**0.700**	**0.700**	0.600	0.467
Email	0.01	**0.867**	–	0.833	0.750	0.800	0.667	0.467	0.667
	0.05	–	**0.700**	0.667	0.500	0.600	0.433	0.433	0.433
Blogs	0.01	–	0.767	**0.867**	0.700	0.800	0.667	0.466	0.200
	0.05	–	0.833	**0.867**	0.567	0.667	0.533	0.433	0.200
CA-HepTh	0.01	–	**0.567**	**0.567**	0.533	**0.567**	0.533	0.433	0.367
	0.05	–	0.667	**0.700**	0.533	0.500	0.500	0.467	0.267
PGP	0.01	–	**0.733**	0.600	0.633	**0.733**	0.700	0.433	0.367
	0.05	–	0.700	**0.750**	0.633	0.600	0.567	0.333	0.333

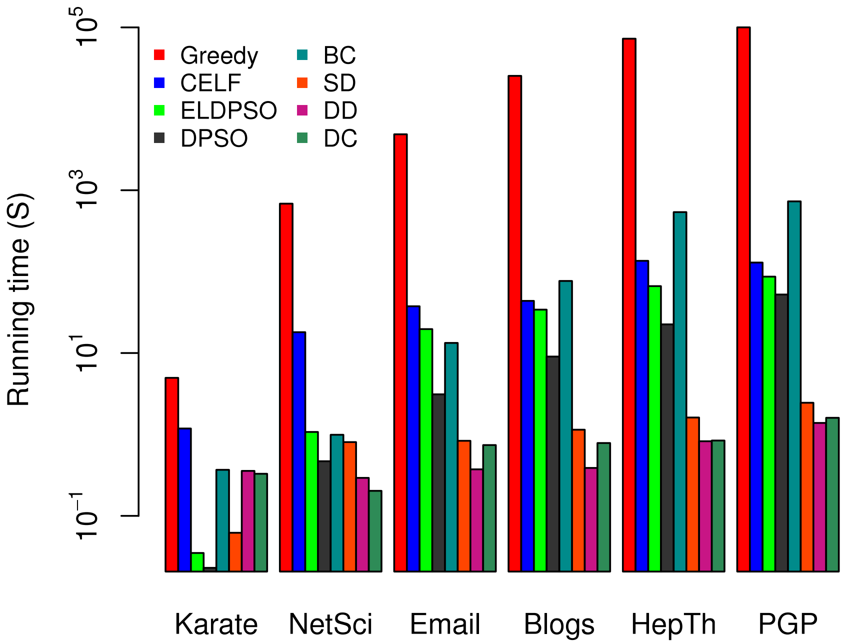


Fig. 5. Comparison on the running time of the eight algorithms when to identify the targeted seed nodes under probability $p = 0.01$ in the six social networks.

From Table 2, we can see that the seed sets achieved by ELDPSO on the six social networks are very approximate to those achieved by greedy algorithm and CELF. In contrast, the original DPSO performs as a mediocre method compared to ELDPSO due to its premature characteristic. What is worse is the centrality-based algorithms, the low ratios indicate that they are unsuitable for multiple influential nodes identification. Meanwhile, further non-parametric tests [51] based on Wilcoxon signed-rank test using SPSS statistics for analyzing the performance of the eight algorithms for influence maximization problem are carried out and given in the second part of Appendix C.

Besides measuring the performance of eight algorithms on influence propagation, comparison on processing time to identify $k = 30$ (for Karate network, $k = 20$) seed nodes under the two propagation probabilities $p = 0.01$ and $p = 0.05$ in the six social networks is also depicted in Figs. 5 and 6 respectively.

From Fig. 4 we know that greedy algorithm is always the best one in identifying the targeted k seed nodes. However, it consumes the highest time in all the six social networks, which proves the conclusion that greedy algorithm is not scalable to large-scale networks when adopted to solve influence maximization problem. CELF is cost-effective in identifying a small subset seed nodes in small-scale networks, but it is incapable in large-scale networks especially when the node degree of the network topology is uniformly distributed. Having to calculate all the shortest paths of each node in the entire network, the global centrality metric BC is inefficient in consideration of the performance on influence maximization, as shown in Fig. 4. In addition to greedy algorithm, CELF and BC, the local centrality DC and SD, as well as the semi-local metric DD, seems to need less time to identify the targeted k seed nodes, but they perform worse than ELDPSO. Compared with DPSO, ELDPSO needs more time for the identification, but it outperforms DPSO in identifying more effective seed nodes as shown in Fig. 4 and is cost-effective and scalable to large-scale networks.

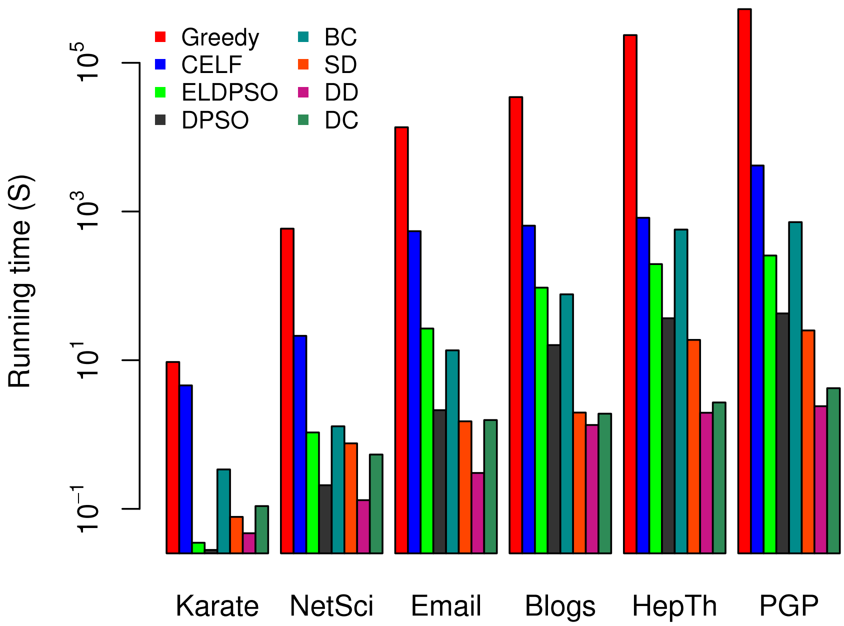


Fig. 6. Comparison on the running time of the eight algorithms when to identify the targeted seed nodes under probability $p = 0.05$ in the six social networks.

5. Conclusions

Influence maximization remains an open problem in the areas including social network analysis and viral marketing. It is necessary to do further research on developing effective algorithms with low time complexity. In this paper, a discrete particle swarm optimization with an enhanced local strategy (ELDPSO) is proposed for influence maximization problem. To avoid the original DPSO being trapped into suboptimal solution easily, we analyze the sensitivity of local search strategy based on network topology for DPSO prudently and present an enhanced topology-based local search strategy. To make use of the local network topology of the seed set, k candidate nodes in the temporary optimal seed set of each generation are recombined in ascending order by the degree centrality metric firstly, and then a local greedy-based strategy is applied on the ordered seed set to exploit more influential nodes. The simulated results show that the enhanced topology-based local search strategy achieves better *LIE* value in different scale of social networks compared with the original DPSO algorithm. The experimental results under IC model prove ELDPSO to be a promising effective and robust method with a low time complexity compared to greedy-based algorithms.

Acknowledgments

This work is supported by the National Natural Science Foundations of China (Grant No. 21503101 and No. 61702240), the Project-sponsored by SRF for ROCS, China, SEM, China (Grant No. SEM[2015]311) and Fundamental Research Funds for the Central Universities, China (Project No. lzujbky-2017-191).

Appendix A. Particle swarm optimization

Particle swarm optimization (PSO) is a population-based memetic-evolution-motivated metaheuristic. The main ideology of PSO can be illustrated as follows: in a d dimensional searching space of the problem to be solved, a population of N particles with respective position and velocity vectors fly to explore the optimal solution sanely, they learn from each other during the searching process and finally converge to the global optimization. Among the state-of-the-art work on PSO, Shi et al. [38] presented a classical weighted PSO (WPSO) and gave the specification of parameters evaluation to assure the WPSO converges to global optimal solution. The mathematical model of WPSO can be formulated as follows:

$$V_i = \omega V_i + c_1 r_1 (Pbest_i - X_i) + c_2 r_2 (Gbest - X_i) \tag{A.1}$$

$$X_i = X_i + V_i \tag{A.2}$$

$$\boxed{1 \mid 34 \mid 12 \mid 6 \mid 33} \cap \boxed{34 \mid 1 \mid 2 \mid 33 \mid 9} \rightarrow \boxed{1 \mid 1 \mid 0 \mid 0 \mid 1}$$

$$\boxed{1 \mid 34 \mid 12 \mid 6 \mid 33} \cap \boxed{34 \mid 1 \mid 33 \mid 12 \mid 2} \rightarrow \boxed{1 \mid 1 \mid 1 \mid 0 \mid 1}$$

Fig. B.7. An illustration of operator "∩".

where $V_i = (v_{i1}, v_{i2}, \ldots, v_{id})$ and $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ represent the position vector and velocity vector of particle $i(i = 1, 2, \ldots, N)$ respectively.

Parameter inertia weight ω in Eq. (A.1) is employed to adjust the impact of the previous historical best velocity of each particle on the current velocity. It means that a larger inertia weight ω facilitates global exploration while a smaller inertia weight ω tends to facilitate local exploitation to fine-tune the current search area. c_1 and c_2 are two positive learning constants, r_1 and r_2 are two random numbers drawn uniformly from $(0, 1)$. $Pbest_i$ denotes the previous historical best position of particle i, and *Gbest* denotes the global best position of the population up to current generation. The second and the third section at the right of Eq. (A.1) are known as 'self-learning' and 'social cognition' factors respectively to adjust the trajectories of the population to global optimal solution.

Appendix B. DPSO for influence maximization

① **Operator** "∩"

Let $X_i = (1, 34, 12, 6, 33)$ be the position vector of particle i, $Pbest_i = (34, 1, 2, 33, 9)$ and $Gbest = (34, 1, 33, 12, 2)$ be the previous historical best position of particle i and the global best position of the swarm separately, if we employ operator "∩" on the both vectors, the illustration can be shown in Fig. B.7.

② **Function** $H(\cdot)$

$H(\cdot)$ is a velocity decisioning function to calculate velocity V_i once the "∩" operation is finished. According to Ref. [32], assuming that the parameter is X_i, $H(X_i)$ can be represented as $H(X_i) = (h_1(x_{i1}), h_2(x_{i2}), \ldots, h_k(x_{ik}))$, where $h_j(x_{ij})(1 \le j \le k)$ is defined as a threshold function formulated as in Eq. (B.1).

$$h_j(x_{ij}) = \begin{cases} 0 & if \;\; x_{ij} < 2 \\ 1 & if \;\; x_{ij} \ge 2 \end{cases} \tag{B.1}$$

For example. if we fix the coefficients $c_1 r_1 = 0.9$ and $c_2 r_2 = 1.3$, then the velocity vector V_i can be calculated as $V_i = H(0.9 * (1, 1, 0, 0, 1) + 1.3 * (1, 1, 1, 0, 1)) = H((2.2, 2.2, 1.3, 0, 2.2)) = (1, 1, 0, 0, 1)$ based on the operation result of ①.

③ **Operator** "⊕"

According to Eq. (6), the operator "⊕" is adopted to judge whether the element in X_i should be kept or adjusted based on the value of V_i formulated by Eq. (B.2).

$$x'_{ij} = \begin{cases} x_{ij}, & if \;\; v_{ij} = 0 \\ Replace(x_{ij}, N), & if \;\; v_{ij} = 1 \end{cases} \tag{B.2}$$

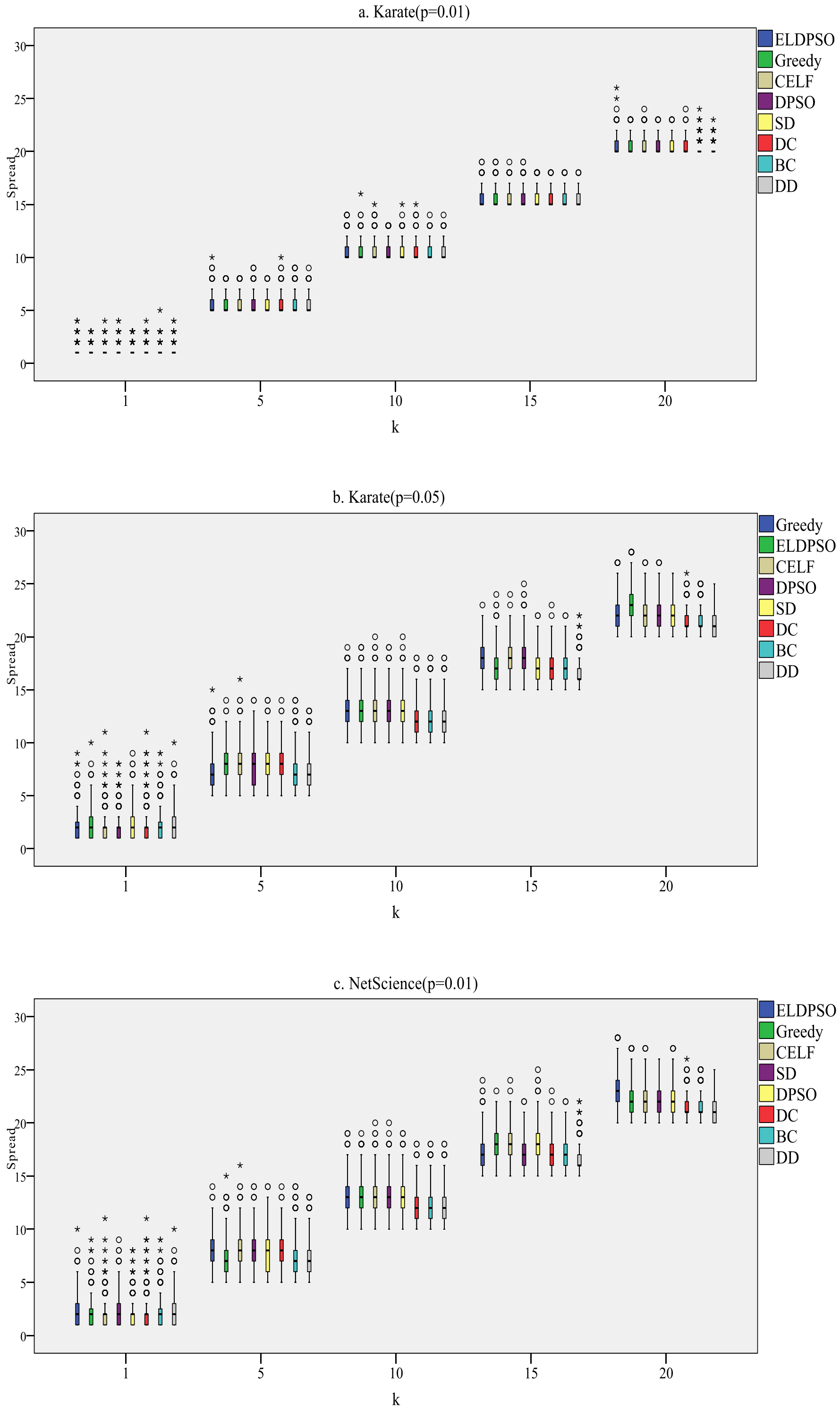
where x_{ij} is an element of X_i, $Replace(\cdot)$ is a function that replaces the element x_{ij} with a random node from the node set N and guarantees there is no repeated node in X_i after the replacement is finished.

Appendix C. Comparison and statistical tests

C.1. Comparison on influence propagation using boxplot

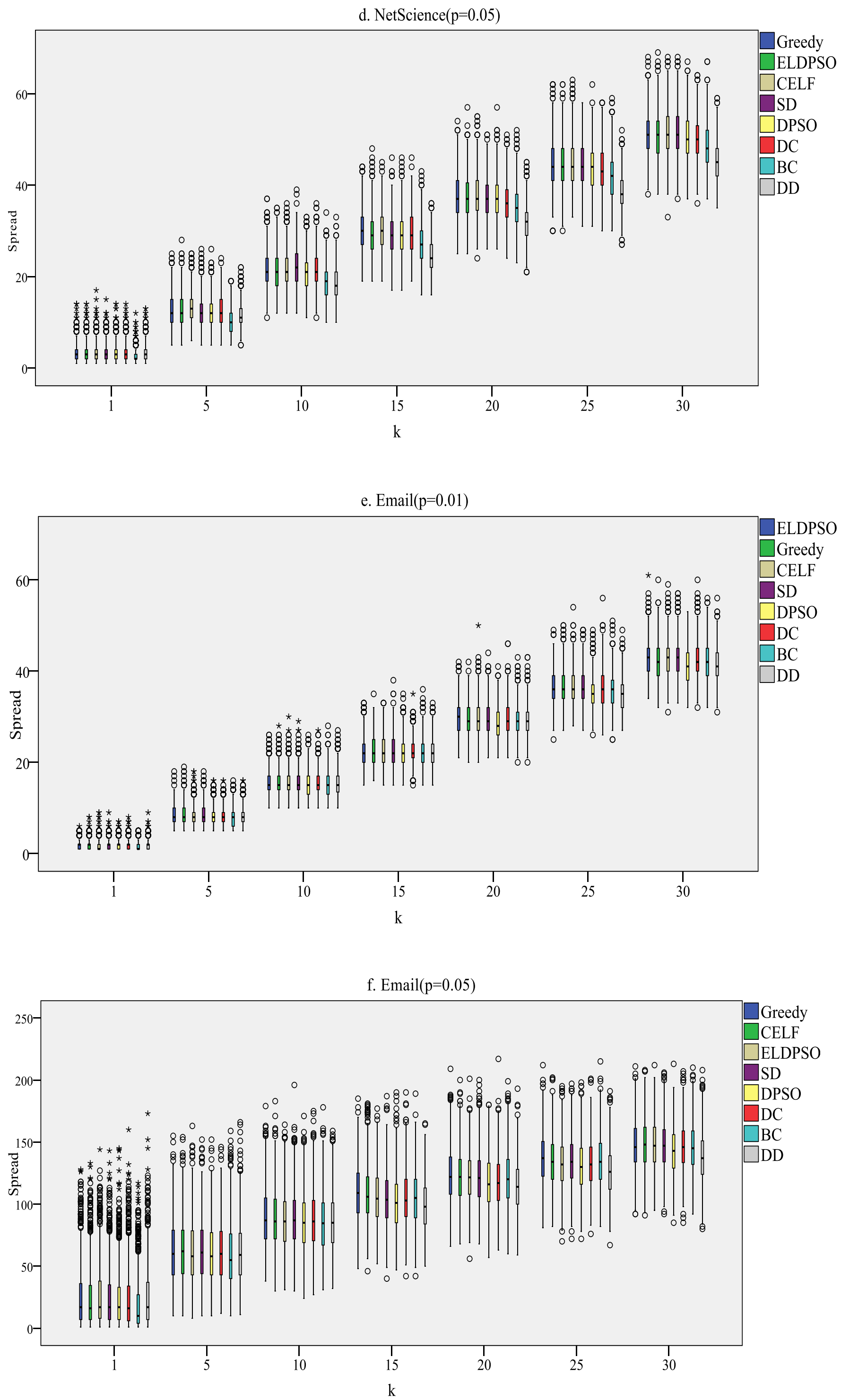
Besides the comparison on the average influence propagation of the eight algorithms under the two probabilities in the six social networks shown in Fig. 4, we also show the simulated results using boxplot to further explore the implicit differences among the eight algorithms by tackling with influence maximization problem in the six social networks.

All valid simulated numbers of influence propagation including special cases of each k scenario are reserved for the further exploration. As shown in Fig. C.8, all the twelve subfigures display almost the same performance as shown in Fig. 4. ELDPSO achieves comparable results to greedy algorithm and CELF. Meanwhile, ELDPSO outperforms DPSO and other heuristics in all the twelve subfigures. In addition, the subfigures in Fig. C.8 also present extra details compared to Fig. 4. In the twelve subfigures in Fig. C.8, each of the eight algorithms has a narrower numerical range of the simulated influence spread and more special cases when k is small, especially in $k = 1$ scenario. With the increasing of k, all the eight algorithms show more stable performance than smaller k cases. Moreover, all the algorithms tend to achieve approximately the same minimum boundary, but variant maximum boundary in the box plots, except Fig. C.8(l), in which the larger average shorted path distance always increase the uncertainty of influence spread.



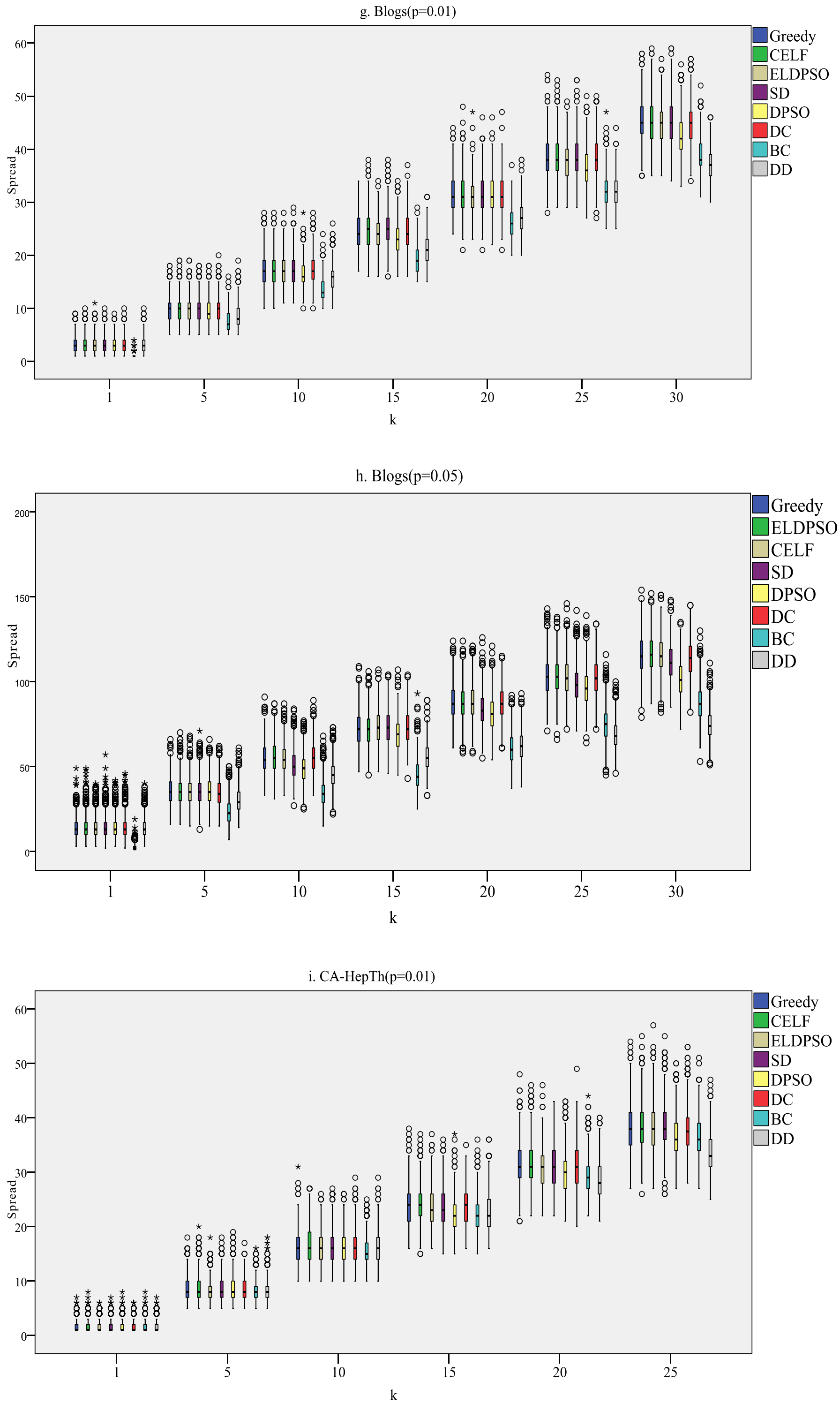
(a)

Fig. C.8. Comparison on the influence spread of the eight algorithms when to select $k = 30$ (for Karate, $k = 20$) seed nodes under propagation probability $p = 0.01$ and $p = 0.05$ in the six social networks.



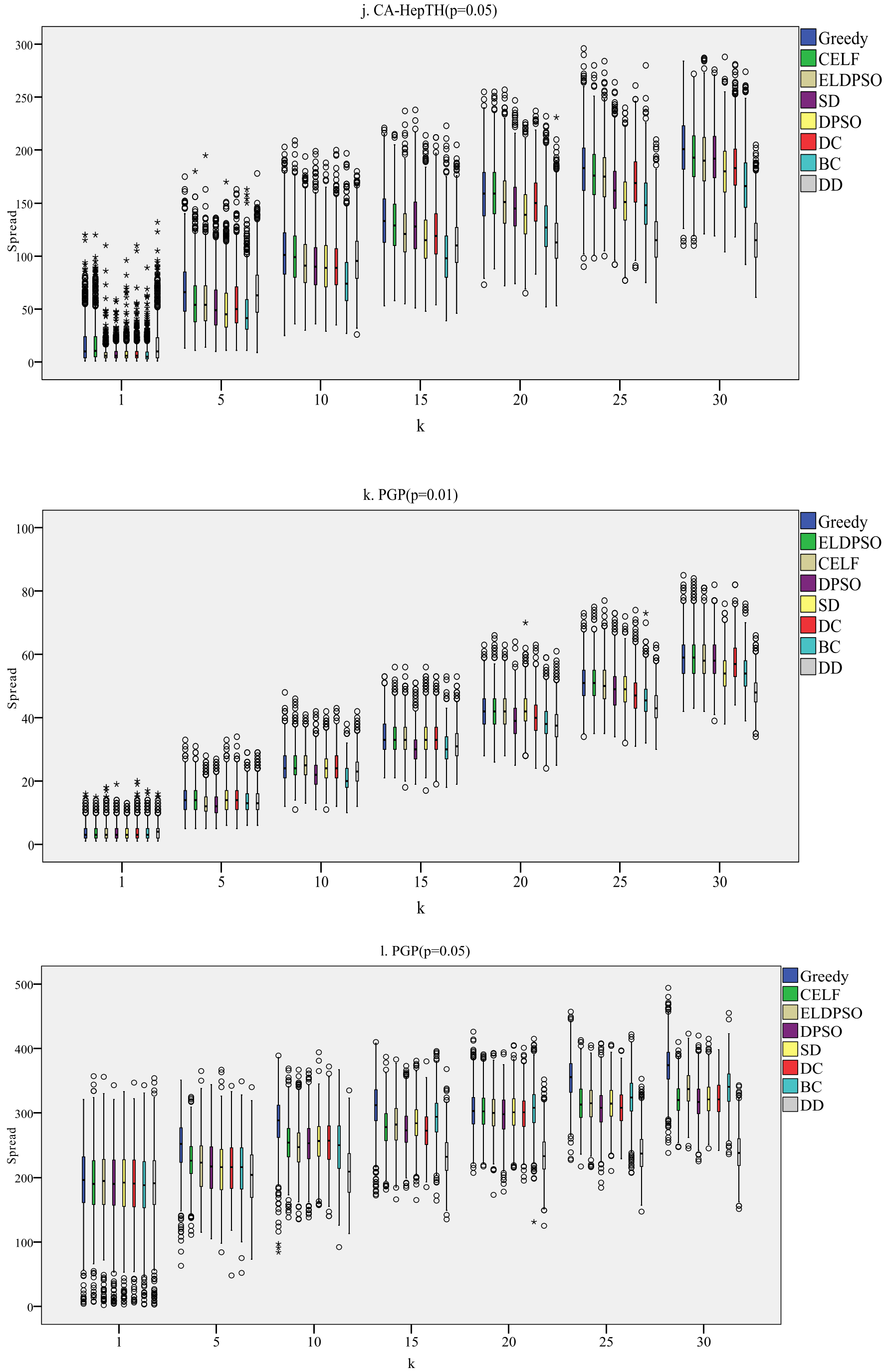
(b)

Fig. C.8. (*continued*).



(c)

Fig. C.8. (*continued*).



(d)

Fig. C.8. (*continued*).

C.2. Statistical tests

To verify the effectiveness of ELDPSO, we conduct a rigorous statistical study in terms of quartile statistics to check whether there is a high level of statistical significance in the results of the eight algorithms on the influence maximization problem in social networks. The multiple-problem Wilcoxons test [52] is performed to check the behaviors of the above eight

Table C.3
Statistical results of the multiple-problem Wilcoxons test for the eight algorithms at $\alpha = 0.05$ and $\alpha = 0.1$ significance level.

ELDPSO	N+	N−	Z	p-value	Adjusted p-value		$\alpha = 0.1$	$\alpha = 0.05$
vs					Holm	Hochberg		
Greedy	3	9	−1.804	0.071	0.142	0.142	No	No
CELF	7	5	−0.235	0.814	0.814	0.814	No	No
DPSO	12	0	−3.059	0.002	0.014	0.006	Yes	Yes
SD	12	0	−3.061	0.002	0.014	0.006	Yes	Yes
DC	12	0	−3.059	0.002	0.014	0.006	Yes	Yes
BC	12	0	−3.059	0.002	0.014	0.006	Yes	Yes
DD	12	0	−3.059	0.002	0.014	0.006	Yes	Yes

algorithms, in which Holm procedure and Hochberg procedure are used as post-hoc procedures. It is necessary to emphasize that multiple-problem Wilcoxons test and post-hoc procedures are accomplished in this paper by using the SPSS software.

Table C.3 summarizes the statistical analysis results, considering ELDPSO as the control algorithm. According to the statistical results, we can see that ELDPSO is significantly better than DPSO and other four heuristics with $\alpha = 0.05$. Meanwhile, it achieves almost the same performance guarantee as the greedy-based CELF algorithm.

References

[1] J. Goldenberg, B. Libai, Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata, academy of marketing science review 9, Mon. Lab. Rev. 31 (3) (2001) 8–11.
[2] Y. Cho, J. Hwang, D. Lee, Identification of effective opinion leaders in the diffusion of technological innovation: A social network approach, Technol. Forecast. Soc. Change 79 (1) (2012) 97–106.
[3] 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.
[4] Y. Li, S. Ma, Y. Zhang, R. Huang, Kinshuk, An improved mix framework for opinion leader identification in online learning communities, Knowl.-Based Syst. 43 (2) (2013) 43–51.
[5] D. Kempe, J. Kleinberg, Tardos, Maximizing the spread of influence through a social network, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137–146.
[6] 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.
[7] C.W.C.J.R. Lee, A query approach for influence maximization on specific users in social networks, IEEE Trans. Knowl. Data Eng. 27 (2) (2015) 340–353.
[8] B.W.B.Z.C. Zhu, Tian Wang, Maximizing the spread of influence ranking in social networks, Inform. Sci. 278 (2014) 535–544.
[9] K.N.R.M.H. Kimura, Masahiro Saito, Extracting influential nodes on a social network for information diffusion, Data Min. Knowl. Discov. 20 (1) (2010) 70–97.
[10] M.F.H. Heidari, Mehdi Asadpour, Smg: Fast scalable greedy algorithm for influence maximization in social networks, Physica A 420 (2015) 124–133.
[11] S.K.P. Suman Kundu, Deprecation based greedy strategy for target set selection in large scale social networks, Inform. Sci. 316 (2015) 107–122.
[12] W.X. Lu, C. Zhou, J. Wu, Big social network influence maximization via recursively estimating influence spread, Knowl.-Based Syst. 113 (2016) 143–154.
[13] S.Y. Wei Chen, Yajun Wang, Efficient influence maximization in social networks, 2009.
[14] J. Shang, S. Zhou, X. Li, L. Liu, H. Wu, CoFIM: A community-based framework for influence maximization on large-scale networks, Knowledge-Based Systems 117 (2017) 88–100.
[15] 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. 415–416 (2017) 171–189.
[16] D. Kim, D. Hyeon, J. Oh, W.S. Han, H. Yu, Influence maximization based on reachability sketches in dynamic graphs, Inform. Sci. 394–395 (2017) 217–231.
[17] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, L.V.S. Lakshmanan, K. Huang, S. Wang, G. Bevilacqua, X. Xiao, L.V.S. Lakshmanan, Revisiting the stop-and-stare algorithms for influence maximization, Proc. VLDB Endowment 10 (9) (2017) 913–924.
[18] H.T. Nguyen, M.T. Thai, T.N. Dinh, Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks, 2017, pp. 695–710.
[19] L. Lš, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, T. Zhou, Vital nodes identification in complex networks, Phys. Rep. 650 (2016) 1–63.
[20] Z.-K. Bao, C. Ma, B.-B. Xiang, H.-F. Zhang, Identification of influential nodes in complex networks: Method from spreading probability viewpoint, Physica A 468 (2017) 391–397.
[21] S. Kundu, C.A. Murthy, S.K. Pal, A new centrality measure for influence maximization in social networks, in: Pattern Recognition and Machine Intelligence - International Conference, Premi 2011, Moscow, Russia, June 27 - July 1, 2011. Proceedings, 2011, pp. 242–247.
[22] M.E.J. Newman, A measure of betweenness centrality based on random walks, Social Networks 27 (1) (2003) 39–54.
[23] L.C. Freeman, Centrality in social networks conceptual clarification, Social Networks 1 (3) (1979) 215–239.
[24] Q. Zhao, H. Lu, Z. Gan, X. Ma, A k-shell decomposition based algorithm for influence maximization.
[25] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (11) (2010) 888–893.
[26] S.H. Antonios Garas, Frank Schweitzer, Shlomo Havlin, A k-shell decomposition method for weighted networks, New J. Phys. 14 (8) (2012) 083030.
[27] T. Bian, Y. Deng, Identifying influential nodes in complex networks: A node information dimension approach, Chaos 28 (4) (2018) 043109.
[28] Y. Saji, M.E. Riffi, A novel discrete bat algorithm for solving the travelling salesman problem, Neural Comput. Appl. 27 (7) (2015) 1–14.
[29] R. Zhang, C. Wu, A hybrid immune simulated annealing algorithm for the job shop scheduling problem, Appl. Soft Comput. 10 (1) (2008) 79–89.
[30] E.P. Ijjina, K.M. Chalavadi, Human Action Recognition using Genetic Algorithms and Convolutional Neural Networks, Elsevier Science Inc., 2016.
[31] 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.
[32] 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.
[33] 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.

[34] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, 1995. Proceedings, Vol. 4, 2002, pp. 1942–1948.
[35] X.S. Yang, A new metaheuristic bat-inspired algorithm, Comput. Knowl. Technol. 284 (2010) 65–74.
[36] J.K.R.C. Eberhart, a new optimizer using particle swarm theory, in: IEEE International Conference on Micro Machine and Human Science, 1995, pp. 39–43.
[37] R.C.E.J. Kennedy, A discrete binary version of the particle swarm algorithm, in: IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, 1997, pp. 4104–4108.
[38] Y. Shi, R. Eberhart, Modified particle swarm optimizer, in: IEEE International Conference on Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, 1999, pp. 69–73.
[39] R. Liu, J. Li, J. Fan, C. Mu, L. Jiao, A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization, Eur. J. Oper. Res. 261 (3) (2017) 1028–1051.
[40] J. Meza, H. Espitia, C. Montenegro, E. Giménez, R. Gonzlez-Crespo, Movpso: Vortex multi-objective particle swarm optimization, Appl. Soft Comput. 52 (2017) 1042–1057.
[41] X. Zhang, X. Zheng, R. Cheng, J. Qiu, Y. Jin, A competitive mechanism based multi-objective particle swarm optimizer with fast convergence, Inform. Sci 427 (2018) 63–76.
[42] F. Zhao, J. Tang, J. Wang, Jonrinaldi, An improved particle swarm optimization with decline disturbance index (ddpso) for multi-objective job-shop scheduling problem, Comput. Oper. Res. 45 (45) (2014) 38–50.
[43] S. Pei, L. Muchnik, J.S.A. Jr, Z. Zheng, H.A. Makse, Searching for superspreaders of information in real-world social media, Sci. Rep. 4 (2014) 5547.
[44] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.
[45] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E: Stat. Nonlinear Soft Matter Phys. 74 (3 Pt 2) (2006) 036104.
[46] R. Guimer, L. Danon, A. Dazguilera, F. Giralt, A. Arenas, Self-similar community structure in a network of human interactions, Phys. Rev. E: Stat. Nonlinear Soft Matter Phys. 68 (2) (2003) 065103.
[47] S. Gregory, Finding Overlapping Communities using Disjoint Community Detection Algorithms, Springer Berlin Heidelberg, 2009.
[48] F. Lu, W. Zhang, L. Shao, X. Jiang, P. Xu, H. Jin, Scalable influence maximization under independent cascade model, J. Network Comput. Appl. 86 (2017) 15–23.
[49] U. Brandes, A faster algorithm for betweenness centrality*, J. Math. Sociol. 25 (2) (2001) 163–177.
[50] Q. Wang, Y. Jin, S. Cheng, T. Yang, Conformrank: A conformity-based rank for finding top-k influential users, Physica A 474 (2017) 39–48.
[51] S. García, 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) (2008) 617.
[52] S. Garca, 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 cec2005 special session on real parameter optimization, J. Heuristics 15 (6) (2009) 617–644.