IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 13, 2021, accepted January 20, 2021, date of publication February 1, 2021, date of current version February 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056087

Identifying Top-k Influential Nodes Based on Discrete Particle Swarm Optimization With Local Neighborhood Degree Centrality

LIHONG HAN[1,4], QINGGUO ZHOU[1], JIANXIN TANG[2], XUHUI YANG[3], AND HENGJUN HUANG[4]

[1]School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China
[2]School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
[3]Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou 730000, China
[4]School of Statistics, Lanzhou University of Finance and Economics, Lanzhou 730020, China

Corresponding author: Qingguo Zhou (zhouqg@lzu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFC0832500, in part by the Ministry of Education—China Mobile Research Foundation under Grant MCM20170206, in part by the Fundamental Research Funds for the Central Universities under Grant lzujbky-2019-kb51 and Grant lzujbky-2018-k12, in part by the National Natural Science Foundation of China under Grant 61402210, in part by the Major National Project of High Resolution Earth Observation System under Grant 30-Y20A34-9010-15/17, in part by the State Grid Corporation of China Science and Technology Project under Grant SGGSKY00WYJS2000062, in part by the Program for New Century Excellent Talents in University under Grant NCET-12-0250, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA03030100, in part by the Google Research Awards and Google Faculty Award, Science and Technology Plan of Qinghai Province under Grant 2020-GX-164, in part by the National Social Science Fund Project under Grant 20XTJ005 and Grant 18BTJ001, in part by the Zhejiang Provincial Natural Science Foundation under Grant LQ20F020011, and in part by NVIDIA Corporation with the donation of the Jetson TX1.

ABSTRACT The top-k influential individuals in a social network under a specific topic play an important role in reality. Identifying top-k influential nodes of a social network is still an open and deeply-felt problem. In recent years, some researchers adopt the swarm intelligence algorithm to solve such problems and obtain competitive results. There are two main algorithm models for swarm intelligence, namely Ant Colony System (ACS) and Particle Swarm Optimization (PSO). The discretized basic Particle Swarm Algorithm (DPSO) shows comparable performance in identifying top-k influential nodes of a social network. However, the performance of the DPSO algorithm is directly related to the choice of its local search strategy. The local search strategy based on the greedy mechanism of the initial DPSO can easily lead to the global suboptimal solution due to the premature convergence of the algorithm. In this paper, we adopt the degree centrality based on different neighbourhoods to enhance its local search ability. Through experiments, we find that local search strategies based on different neighbourhoods have significant differences in the improvement of the algorithm's global exploration capabilities, and the enhancement of the DPSO algorithm based on the degree centrality of different neighbourhoods has a saturation effect. Finally, based on the degree centrality of the best neighbourhood with improved local search ability, we propose the DPSO_NDC algorithm. Experimental results in six real-world social networks show that the proposed algorithm outperforms the initial DPSO algorithm and other state-of-the-art algorithms in identifying the top-k influence nodes.

INDEX TERMS Discrete particle swarm optimization, local search strategy, neighbourhood degree centrality, top-k influential nodes, social network.

I. INTRODUCTION

People from all sides of the world can connect and form a specific complex network, such as wireless sensor network [1], traffic network [2], software network [3] and so on. With the development of information technology, the scale of a social network is becoming larger and larger, and it becomes more and more impractical to make a comprehensive analysis of such a social network. For a specific topic in a social network, identifying the top-k influential nodes is a crucial issue, which relates to the wide applications, such as information spreading [4], [5], management and control [6]–[8], viral

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Zakarya.

marketing [9], [10], traffic bottlenecks identification [11] etc. Up to now, there are many methods to identify the top-k influential nodes in a social network, such as centrality of degree [12], path [13], information entropy [14], coritivity [15], evidence theory centrality [16], k-shell [17] etc. With the increasing scale of the social network, the efficiency of the traditional centrality algorithm is very low. Some heuristic approaches show higher efficiency in solving this problem, such as SAEDV [18],DPSO [19] etc.

In this paper, we proposed a method to identity the top-k influential nodes in a large-scale social network by integrating the DPSO algorithm with the neighbourhood degree centrality of a network. The main contributions of this paper are summarized as follows:

1) We proposed the DPSO_NDC algorithm, whose local search strategy that based on the neighbourhood centrality can enhance the local search ability and can obtain the optimal global solution.

2) By comparing the LIE values of the algorithm with an enhanced local search strategy that based on different neighbouring domains, we find that the neighbourhood degree centrality within the range of 3-hop improves the performance of the initial DPSO algorithm significantly, and followed by 2-hop. However, the performance improvement of the neighbourhood degree centrality within the 4-hop range is far less than that of the 2-hop and 3-hop, or even far worse than 1-hop. In other words, the neighbourhood degree centrality has a saturation effect in improving the global searching capability of the initial DPSO algorithm.

3) Experimental results in the six real social networks show that the proposed DPSO_NDC algorithm has competitive performance with the algorithms proposed in state of the art. Compared with the original DPSO algorithm, its performance is improved significantly.

The rest of the paper is structured as follows: Section 2 reviews some relevant works. The problem definition and the introduction of discrete Particle Swarm Optimization Algorithm is presented in Section 3. The method that integrated the DPSO algorithm with the local neighbourhood centrality is proposed in Section 4. In section 5, the experimental results and comparisons are shown. Finally, the conclusions are summarized in section 6.

II. RELATED WORK

Since Kempe [20] formalized such problem as an optimization problem in 2003, a large number of algorithms have emerged to solve this problem in the social network. In summary, these methods are mainly summarized as the methods based on network structure and heuristic methods. In the research of algorithms based on network structure, Chen *et al.* [21] proposed the Single Discount (SD) algorithm, which sorts all nodes of a network based on degree and iteratively selects a node with the maximum degree. Kundu *et al.* [22] proposed an algorithm based on the diffusion degree of nodes, which filters out the top-k influential individuals with the highest diffusion degree value to form the influential seed node-set. Chen *et al.* [23] introduced path diversity and found that it can largely improve the identifying accuracy. Zhao *et al.* [24] provides GIN (global importance of each node) model to identify influential nodes from the global perspective of the complex networks. Liu *et al.* [25] proposed a GMM model, which combines the local information and global information of the network to identify the most influential nodes in the social network. Wen *et al.* [26] proposed multi-local dimension (MLD) method to identify the vital spreader in the social network and found that the node with low MLD value would be more important in the network. On the other hand, heuristic algorithms based on greedy mechanisms mainly include CELF [27], CELF++ [28], etc. Salehi and Masoumi [29] proposed a metaheuristic algorithm based on Katz centrality to solve IM problem in the social network.

In recent years, swarm intelligence algorithms are used to identify the top-k individuals in a social network. Jiang *et al.* [18] applied the Simulated Annealing (SA) algorithm to obtain the optimal solution of influence maximization. Simsek and Kara [30] employed the Grey Wolf Optimizer (GWO) and Whale Optimization Algorithm (WOA) to solve the influence maximization problem, in which experimental results show that the swarm intelligence approach is effective and efficient. Sankar *et al.* [31] adopted the bee algorithm to explore the bee colony's waggle dance behavior for identifying influential individuals. Tang *et al.* [32] proposed a discrete shuffled frog-leaping algorithm to select top-k influential nodes in a social network. Zareie *et al.* [33] use gray wolf optimization algorithm to identify the influential users in the social network. Singh *et al.* [34] proposed ACO-IM algorithm using the ant colony optimization to solve the maximizing influence in the social network. Sheikhahmadi and Zareie [35] adopted multi-objective artificial bee colony optimization to identify the influential spreaders in a social network. As can be seen from the above-mentioned researches, the application of swarm intelligence algorithm to solve this problem has been paid more and more attention by scholars.

III. PRELIMINARIES

A. PROBLEM DEFINTION

Let $G = (V, E)$ denote a social network graph, where $\{v_1, v_2, \cdots, v_n\}$ denotes a set of nodes and $E = \{e_1, e_2, \cdots, e_m\}$ denotes a set of edges. A node represents an individual actor in a social network graph, and an edge represents the relationship between individuals, such as collaboration, friendship, or certain social relationships. The different social relationships can build different network graphs. To study the influence spread, a node's state in the network models is defined as *active* and *inactive*. The activated node indicates that the corresponding individual can spread information or influence. In contrast, the inactive node denotes that the active nodes can influence the corresponding individual.

Definition 1 (Top-k Influence Nodes): The top-k influence nodes are defined as a set of nodes of size k that have the most influence in the network for a particular topic. It aims at identifying a set of nodes of size k based on a specific topic or a model whose influence or importance in the social network is greater than that of any other nodes outside the set. The significance of node v under a specific mode or topic is denoted as $f(v)$, so the influence node set S' and the relationship between top-k influence nodes can be expressed as in (1):

$$S' = \sum_{v=1}^{k} f(v)\{|S'| = k, f(1) \geq f(2) \geq \cdots \geq f(v) \cdots \geq f(k)\}. \tag{1}$$

where k is the user-specified number of nodes.

Definition 2 (Influence Maximizing Nodes): A set of nodes consisting of K nodes that have the widest range of influence under a given propagation model and propagation probability, i.e. the number of nodes expected to be influenced is the largest. Formally, it is denoted as $d(S)$ [20]:

$$S^* = argmax[d(S)]\{S \in V, |S| = k\}. \tag{2}$$

where k is the size of the seed node-set. It aims at identifying the top-k nodes set of maximum influence spreading. It is an optimization problem and a sub-problem of the identification of the top-k influence nodes.

Definition 3 (Independent Cascade Diffusion Model): The Independent Cascade (IC) model [20] is a random process in a social network. In the IC model, the entire influence propagation process is composed of discrete random propagation steps. In each time step, there are only two states of nodes in the network: *active* state and *inactive* state. In each step of propagation, the node in the active state activates its direct neighbour nodes with probability p, and the node can only change from the inactive state to the active state, and vice versa. For simplicity, the probability pi is usually defined as a constant value.

B. LOCAL INFLUENCE EVALUATOR

The computation of the influence spread $d(S)$ in the social network has been proved to be a NP-hard problem. To solve this problem, Jiang *et al.* [18] proposed a fast EDV objective function that applies to evaluate the spreading scale for the social networks. For the convenience of calculation, Gong *et al.* [19] proposed the Local Influence Estimation (LIE) function to approximate the influence of the nodes. This objective function is the suite for the influence spread approximation of large-scale social networks in the IC model. The estimation function LIE is formulated as follows:

$$\begin{aligned} LIE &= \sigma_0(S) + \sigma_1^*(S) + \sigma_2^{\sim}(S) \\ &= k + \sigma_1^*(S) + \frac{\sigma_1^*(S)}{|N_S^{(1)} \setminus S|} \sum_{u \in N_s^{(2)} \setminus S} p_u^* d_u^* \\ &= k + (\frac{1}{|N_S^{(1)} \setminus S|} \sum_{u \in N_s^{(2)} \setminus S} p_u^* d_u^*) \\ &\quad \cdot \sum_{i \in N_S^{(1)} \setminus S} (1 - \prod_{(i,j) \in E, j \in S} (1 - p_{i,j})) \end{aligned} \tag{3}$$

where $N_s^{(1)}$ and $N_s^{(2)}$) are the sums of the degree values within the range of one-hop and two-hop of each node in the seed node set S, respectively. $Pu*$ is the constant probability of successfully activating its neighbour node. $Du*$ is the number of edges for node u within the range of one-hop and two-hop. It has been proved in some literature [36] that the LIE evaluation function has good approximate results. In this way, the problem of identifying the top-k influence nodes becomes an optimization problem of the objective function LIE, that is, the optimal k nodes are selected to form the seed node-set by the maximization principle of the LIE function value.

C. PSO AND DPSO

The basic particle swarm optimization (PSO) algorithm is a classical optimization algorithm, which is inspired by the evolutionary behavior of birds' collective flight. The original PSO model can be formulated as follows:

$$V_i^{t+1} = wV_i^t + c_1 r_1 (Pbest_i - X_i^t) + c_2 r_2 (Gbest - X_i^t) \tag{4}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{5}$$

where $V_i(v_{i1}, v_{i2}, \cdots, v_{id})$ denotes the position vectors of particle i, and $X_i(x_{i1}, x_{i2}, \cdots, x_{id})$ denotes the velocity vector of particle $i(i = 1, 2, \cdots, N)$. N is the size of particle swarm, and d is the dimension of the problem space. Constant learning factors are c_1 and c_2, r_1 and r_2 are two random factors that take the random value from (0, 1). In the wV_i^t item, w is the weight. $Pbest_i = (pbest_{i1}, pbest_{i2}, \cdots, pbest_{ik})$ denotes the optimal position of particle i, and $Gbest = (gbest_1, gbest_2, \cdots, gbest_k)$ represents the global optimal position in the particle swarm.

Above traditional PSO is suitable for the solution space of continuous function and has good optimization performance. To apply the basic PSO algorithm to the discrete complex network space, Gong *et al.* [19] proposed discrete particle swarm optimization (DPSO) algorithm based on redefining the form of the basic PSO. The form of DPSO model is formulated as follows:

$$V_i^{t+1} = H[\omega V_i^t + c_1 r_1 (Pb_i \cap X_i^t) + c_2 r_2 (Gb \cap X_i^t)] \tag{6}$$

$$V_i^{t+1} = X_i^t \oplus V_i^{t+1} \tag{7}$$

where X_i is the integer number of the node ID, and the Pb_i and Gb are vectors composed of X_i. Operator "$H(\cdot)$" is a decision function, operator " $\cap$ " is a logical similar intersection operation. Velocity vector $V_i = (0, 1)$, where 0 indicates that the node corresponding to the location in the current seed node-set X_i does not need to be replaced, 1 represents the node corresponding to the location of seed node X_i needs to be adjusted.

IV. METHOD

In this section, the local search strategy of the DPSO algorithm is discussed and the neighborhood centrality is introduced. Then, we present the improved local search strategy based on the neighbourhood degree centrality to enhance the global searching ability of the initial DPSO algorithm.

A. LOCAL SEARCH STRATEGY ANALYSIS

As the general slope required for the optimization problem cannot be directly generated in the discrete network solution space, the basic DPSO algorithm cannot be directly applied to solving the top-k influence nodes identification in the discrete network space. In the continuous solution space, the basic PSO algorithm follows the general slope over the surface to converge at the global optimum. However, In the IM problem, it cannot converge to the global optimum. To make the particles converge to a global optimal solution in the DPSO algorithm, the author [19] proposed the local search operator that is based on a greedy strategy. For the detailed local search strategy of the DPSO algorithm see Algorithm 1. Where function $Replace(\cdot)$ is used to replace node x_{bi} with one of its nearest neighbor nodes randomly.

Algorithm 1 Pseudocode of Local Search Strategy

Input: Particle Xa
Output: Particle Xb

1: $Xb \leftarrow Xa$
2: **for** $xb_i \in Xb$ **do**
3: $Flag \leftarrow False$
4: $Neighbors \leftarrow N_{xbi}(1)$
5: **repeat**:
6: $xb_i \leftarrow Replace(xb_i, Neighbors)$
7: **if** $LIE(Xb) > LIE(Xa)$ **then**
8: $Xa \leftarrow Xb$
9: **else**
10: $Flag \leftarrow True$
11: **until** $Flag == True$
12: $Xb \leftarrow Xa$
13: **return** Xb

From the local search strategy of DPSO, it can be seen that if the replacement operation is executed iteratively, i.e., $LIE(Xb) > LIE(Xa)$, it may return an expected optimal result. However, if the replacement operation is executed prematurely to meet the condition of termination (i.e., in statement on line 10), it will fall into a suboptimal solution. Therefore, the global optimal solution of the DPSO algorithm has a crucial relationship with the local search strategy based on the greedy mechanism; it can effectively solve the problem of global optimal convergence to some extent. However, the local search strategy of DPSO is still easy to make the algorithm premature and fall into the suboptimal solution.

B. PROPOSED ALGORITHM DPSO_NDC

As a meta-heuristic algorithm, DPSO has great advantages in the optimization of a large-scale network, and the global optimal solution of DPSO is closely related to the local search strategy. Different local search strategies directly affect the performance of DPSO algorithm.

1) NEIGHBORHOOD CENTRALITY

Among the identifying methods of top-k influential nodes, although the centrality-based methods are simple, its calculation cost is very high and it is not suitable for the large-scale network structure. Chen *et al.* [23] designed a semi-local centrality method to identify influential nodes, which is a tradeoff between efficiency and complexity of the centrality-based approach. Liu *et al.* [37] proposed a way to measure the influence of nodes, which is called neighbourhood centrality that based on the centrality of a node itself and its nearest neighbors.

Based on the inspiration of the above two works, we make full use of this feature of the network local topology to improve the local search ability of the DPSO algorithm. We defined the neighbourhood centrality (NDC) of a node as follows:

$$C_i^n(\theta) = \sum_{j \in \Gamma_i} \theta_j + \sum_{l \in \Gamma_j \setminus i} \theta_l + \cdots + \sum_{s \in \Gamma_{s-1} \setminus z} \theta_s \qquad (8)$$

where θ is the metric of benchmark centrality, and Γ_i is the node-set of nearest neighbours of node i. The number of items in Eq.(8) is the hop of neighbours that taken into consideration. This equation means that the neighbourhood centrality contains the centrality of a node itself and its neighbour's area of considering hop.

2) LOCAL SEARCH STRATEGY WITH NDC

For the calculation of the neighbourhood degree centrality, we calculate up to the fifth items in Eq.(8) to enhance the local search capability for the proposed local search strategy. The pseudo code of the improved local search strategy with neighbourhood degree centrality is given in Algorithm 2. The set of temporary seed nodes in each iteration is sorted in ascending order of their neighbourhood degree centrality. In this way, the node in the seed node-set with low neighbourhood degree centrality is given priority to search that avoids the trap of suboptimal solution when meeting the termination condition prematurely. Besides, the greedy selection strategy is used to replace the current optimal seed node, that is, the node with the largest LIE value is selected to displace the current optimal temporary seed node from the set of its neighbourhood node set. In Algorithm 2, function $SumNBDegree(gbest_i)$ calculates the degree of the x-step area of node $gbest_i$. Function $AscSor(NBDSet_{xstep})$ sorts nodes in the node-set $NBDSet_{xstep}$ by ascending order of degree. Function $Size(Neiborhoodset)$ returns the size of the node-set $Neibourhoodset$, and function $Replace(gbest_i, NeiborhoodSet)$ is used to replace the node

Algorithm 2 Local Search Strategy Based on Neighborhood Degree Centrality

Input: $Gbest^*$, //globally optimal node set of current iteration.
Output: $Gbest^*$, //The globally optimal seed set.

$NBD_{Gbest^*} \leftarrow Gbest^*$
for each element $gbest_i \in NBD_{Gbest^*}$ **do**
  $Flag \leftarrow False$
  $NBDSet_{xstep} \leftarrow SumNBDegree(gbest_i)$
  $NeiborhoodSet \leftarrow AscSort(NBDSet_{xstep})$
  $index \leftarrow 0$
  while $index < size(NeiborhoodSet)$ **do**
    $gbest_i \leftarrow Replace(gbest_i, NeiborhoodSet)$
    if $LIE(NBD_{Gbest^*}) > LIE(Gbest^*)$ **then**
      $Gbest^* \leftarrow NBD_{Gbest^*}$
      break
    else
      $index \leftarrow index + 1$
return $Gbest^*$

$gbest_i$ with one of the x-hop area neighbours and excludes the duplicate notes.

To observe the improvement effect of neighbourhood degree centrality of different hop areas on the local search strategy, function *SumNBDegree*() realizes the calculation of X-hop area degree centrality of seed node set. The pseudo code of function *SumNBDegree*() is presented in Algorithm 3.

Algorithm 3 Calculation of Neighbourhood Degree Centrality

Input: *XStep*, //The step areas.
$gbest_i$, //the Global best seed node i.
Output: *Nodeset*, //The XStep area node set.

$Neighbors \leftarrow N^{(1)}_{gbest_i}$
for each element $x_{nbi} \in Neighbors$ **do**
  $Step \leftarrow 1$
  repeat:
    $Nodeset \leftarrow Neighbors^{step}_{x_{nbi}}$
    $Step \leftarrow Step + 1$
  until $Step > XStep$
  $Nodeset \leftarrow RemoveDuplicates(Nodeset)$
  $Nodeset \leftarrow DifferenceSet(Nodeset, Gbest)$
return *Nodeset*

3) THE FRAMEWORK OF DPSO_NDC

From the above analysis, it can be seen that the local search strategy plays a crucial role in obtaining the optimal solution. A proper local search operation is helpful to the exploration of a global optimal solution for the algorithm. The algorithm DPSO_NDC, which uses the above-defined neighbourhood centrality to enhance the local search ability, has a framework as described in the pseudo code of Algorithm 4. In the framework of DPSO_NDC, the velocity vector and position vector

Algorithm 4 Framework of DPSO_NBC for Identifying Top-k Influence Nodes

Input: $G = (V, E)$, //Graph.
  g_{max}, //the number of iterations.
  c_1, c_2, //the learn factors.
  k, //the size of the seed set.
  w, //the inertia weight.
Output: $Gbest^*$, //The global best position as the seed set.

1: initialize position vector X based on degree centrality.
2: initialize velocity vector V to 0.
3: Select out the interim best solution Gbest based on the LIE value of vector X.
4: **repeat**:
5: Update the velocity vector V based on Eq.(5).
6: Update the position vector X based on Eq.(6).
7: Update the Pbest and select out the current global best particle Gbest.
8: Calculation the X-hop area degree centrality of each particle in current interim best particle solution set Gbest based on Eq.(6).
9: $Gbest \leftarrow LocalSearch(Gbest^*)$// Employ the improved local search operation on Gbest* based on X-hop area degree centrality.
10: $Gbest^* \leftarrow Max(Gbest)$ // Update the Gbest*.
11: Next iteration.
12: **until** Up to the maximum number of iterations.
13: **return** $Gbest^*$

of each generation are updated to find the current global optimal solution firstly, as described in statement on lines 5-7. Then, the neighbourhood degree centrality of the specified domain range of each node in the current temporary global optimal solution is calculated, as described in statement on line 8. Once the optimal seed node set in current evolution is obtained, we adopt the improved local search strategy to explore the current optimal seed node locally and find better candidate seed nodes for next-generation evolution until the termination condition is satisfied, as described in statement on line 9.

4) COMPUTATIONAL COMPLEXITY ANALYSIS

The computational complexity is a considerable metric to evaluate the performance of the algorithm. Compared with the initial DPSO algorithm, the computational complexity of algorithm DPSO_NDC is different mainly in the local search strategy. For the improved local search operation, the computation time mainly lies in the computation of degree centrality of neighbourhood area and the ranking operation of temporary seed nodes. The computation of x-hop neighbourhood degree centrality requires $O(k \cdot D^x)$, and the ordering operation requires $O(k \cdot logk)$. According to the framework of DPSO_NDC, the other operating complexity is as follows: the complexity of updating velocity is $O(k \cdot logk \cdot N)$; the complexity of updating position X is $O(k \cdot N)$; evaluating LIE

value needs $O(k \cdot D^2)$; the greedy-based replacement needs $O(k \cdot D)$. Thus, the computational complexity of DPSO_NBC requires $O(K^2 \cdot logk \cdot N \cdot D^x \cdot gmax)$. It can be seen that the computational complexity of this algorithm increases exponentially with the range of the nearest neighbour domain. Therefore, it is very important to select an appropriate nearest neighbour domain between the computational complexity and the optimal performance.

5) EVALUATION AND COMPARISONS

In this section, the experiments in the six real-world social networks are carried out to verify the performance of algorithm DPSO_NBC. At the same time, we selected the 5 algorithms in the state of the art as benchmark algorithms and compared their fitness values. Also, we used the Monte Carlo (MC) method to simulate the propagation scale of the seed nodes of these algorithms under the best fitness value, and the scale of their simulated propagation is compared.

C. EXPERIMENTAL NETWORKS AND BASELINE ALGORITHMS

1) EXPERIMENTAL NETWORKS

To make verification and comparison more objective, we conduct extensive experiments on six real-world undirected social networks. The statistical characteristics of the six experimental networks are listed in Table 1.

TABLE 1. Statistical characteristic of the six real-world networks.

Networks	|V|	|E|	<K>	<d>	<C>	D
NetScience [38]	379	914	4.824	6.042	0.798	0.013
Email [39]	1133	5452	9.624	3.606	0.254	0.009
HepTH [40]	9877	25998	5.26	5.945	0.601	0.001
PGP [41]	10680	24316	4.554	7.486	0.44	0.001
GrQc [42]	5242	28980	11.54	6.049	0.687	0.001
CondMat [43]	23133	186936	16.156	5.353	0.706	0.001

|V| is the number of nodes, |E| is the number of edges. <K> represents the average degree of the network, <d> represents the average path distance, <C> is the average clustering coefficient of the network, D is the density of the network.

Social network PGP [41] is generated by the Pretty Good Privacy encryption algorithm. It is consisted of 10680 nodes that represent the people who share confidential information. Social network GrQc [42] is a collaboration network of the Arxiv General Relativity category, where the edges represent co-authored relationships between different authors in the same article. The all of experimental social network are get from SNAP1. The node degree distributions of each experimental networks are as shown in Figure.1. The experiments consist of two parts. First, the comparisons of LIE value between DPSO and DPSO_NDC are conducted in the six real-world networks. In addition, the four other state-of-the-art algorithms are employed as baseline algorithms, and the influence spread performance of the identified top-k nodes is simulated for five algorithms, respectively.

2) BASELINE ALGORITHMS AND EXPERIMENTS

The five other state-of-the-art algorithms are employed as baseline algorithms and the influence spread performance is simulated under the IC diffusion model for six algorithms, respectively. The five baseline algorithms for comparison are as described in Table 2.

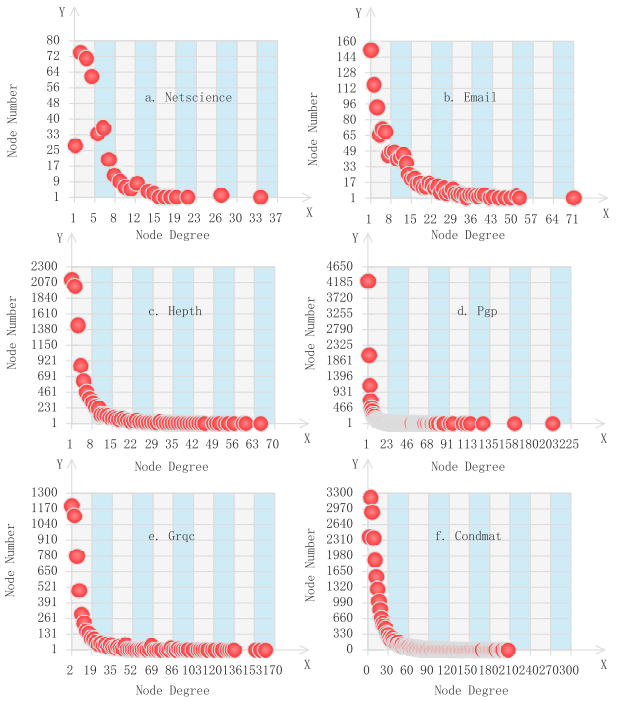

FIGURE 1. The degree distribution characteristic chart of six experimental social networks.(It can be seen from the form of the degree distribution that the real networks of the six experiments all conform to the typical power-law distribution.)

TABLE 2. The baseline algorithms and their brief overview.

Algorithm	Brief Overview
DBA [36]	Discrete Bat Algorithm (DBA) is a discrete meta-heuristic algorithm, which derives from the basic bat algorithm and is used to identify the most influential node sets in social networks.
DPSO [19]	Discrete Particle Swarm Optimization(DPSO) is a metaheuristic algorithm which discretizes the basic Particle Swarm Optimization (PSO) algorithm. The basic PSO algorithm minis the foraging behavior of flocks and is a random and parallel optimization algorithm.
Greedy [20]	Greedy strategy is adopted to select the node u with the maximum marginal return and add the u to seed set S in each iteration. Because a lot of iterations are needed in the process of calculating the propagation of influence, the efficiency of the algorithm is very low.
SD [21]	Single Discount (SD) algorithm is a simple degree discount heuristic algorithm, which is a modified degree centrality algorithm where each neighbor of a newly selected node discounts its degree by one.
ELDPSO [44]	Enhanced Local DPSO (ELDPSO) is an improved DPSO algorithm. It enhances the local search capability of the original DPSO algorithm by sorting the candidate seed nodes in ascending order of degree.

The design and process of the experiment are listed below:

- Experimental software environment. The above benchmark algorithms are all re-coded in the C++ language to eliminate the difference in processing time caused by the efficiency of the encoding language. In addition, no other applications are running on the platform during each experiment to ensure that all experiments are conducted under consistent software and hardware conditions.
- Experimental hardware environment. The experiment is conducted on a Server platform equipped with an intel Xeon E5 CPU, and 16GB RAM, with Windows Server 2003 (Service Pack 2) OS installed.
- Experimental design. The experimental simulation is based on the IC model. Experiments were carried out for seed node sets of different scales. We set seed nodes as 3, 5, 10, 15, 20, 30, 40 and 50 respectively. The maximum seed node set is set to 50.
- Experimental procedure. All the experiments are repeated 20 times and the average fitness value is taken for comparison. Then, the seed node set corresponding to the optimal LIE value in the 20 experiments was selected for the MC simulation propagation experiment. In the MC experiment, we performed 1000 MC simulations on the seed node set, and then took the average value as the MC propagation range for comparison.

3) COMPARISON OF LIE VALUE

Due to the consideration of computational cost, we only designed 5 algorithms with 0-hop to 4-hop neighbourhood degree centrality to enhance the local search ability for experiments, where 0-hop is the degree centrality of the current temporary seed node itself. According to the definition of Eq.(8), we express the above neighborhood degree centrality within different hop areas as NDC0, NDC1, NDC2, NDC3 and NDC4, respectively. The corresponding algorithms are expressed as DPSO_NDC0, DPSO_NDC1, DPSO_NDC2, DPSO_NDC3 and DPSO_NDC4, respectively. To verify the improved performance of local search strategy based on neighbourhood degree centrality of the different ranges, the experimental comparison is made among the improved local search ability with different hop domains. For the parameter setting of these algorithms, we adopted the most effective parameter value evidenced by the initial DPSO algorithm in Ref [19], i.e., the learning factors c_1 and c_2 are set to 2, and the inertia weight w is set to 0.8. When the propagation probability p is set to 0.01, the evolutionary line graph of LIE values is shown in Figure 2. When the propagation probability p is set to 0.05, the spreading scale of evolutionary processes is shown in Figure 3.

From Figure 2, we can observe that the LIE value does not change much when seed set size $k < 30$. However, when the seed set size $k > 30$, the difference

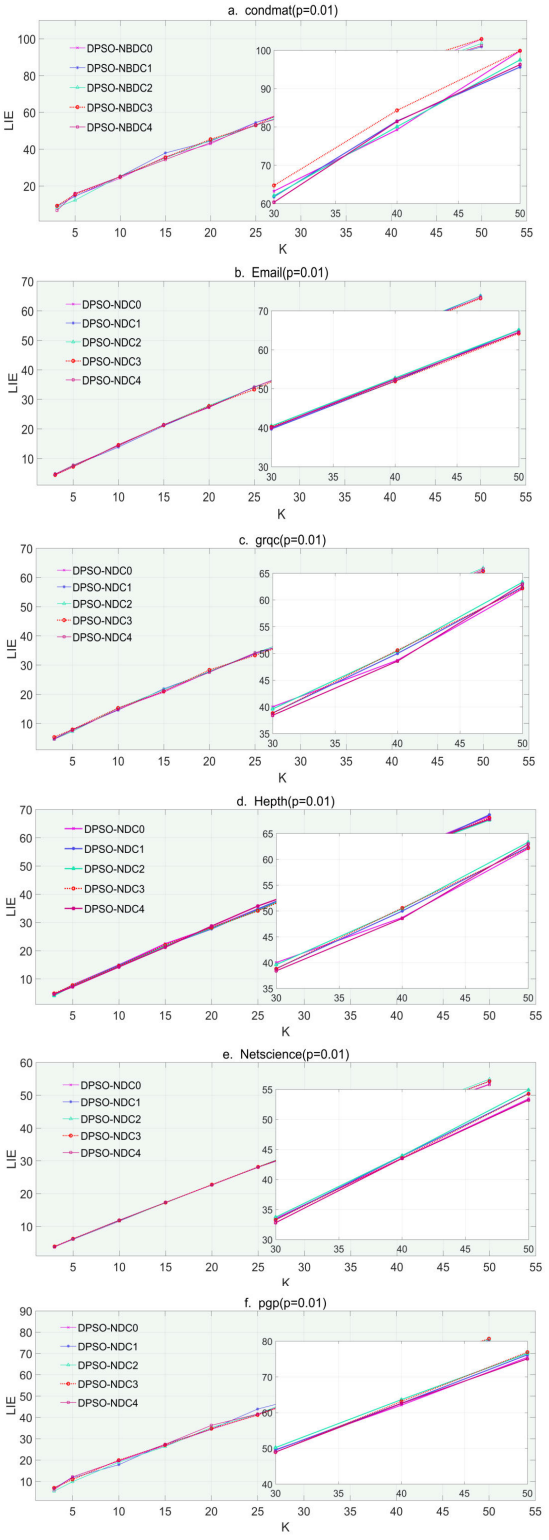

FIGURE 2. The evolutionary line graph of LIE values in the case of propagation probability p=0.01.(On the horizontal axis, *K* represents the number of seed nodes; on the vertical axis, LIE represents the fitness value corresponding to the seed node-set.The subgraph embedded in each network line graph is a partially enlarged view when the seed node is set to 30-50.)

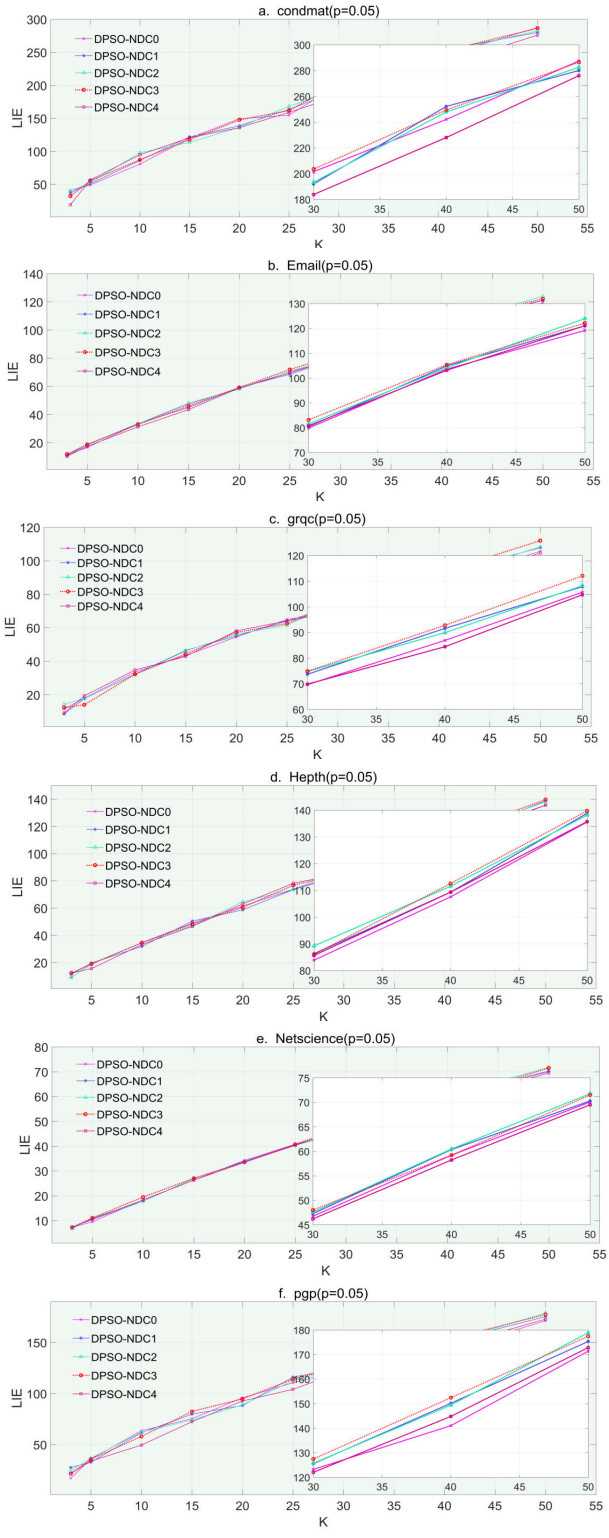

FIGURE 3. The evolutionary line graph of LIE values in the case of propagation probability p=0.05.(On the horizontal axis, *K* represents the number of seed nodes; on the vertical axis, LIE represents the fitness value corresponding to the seed node-set.The subgraph embedded in each network line graph is a partially enlarged view when the seed node is set to 30-50.)

between LIE values is relatively large. Among the above-mentioned six networks, in Condmat and Pgp networks, the DPSO_NDC3 performs best; For network Email, Grqc and Netscience, DPSO_NDC2 and DPSO_NDC3 have similar performance. In network Hepth, the DPSO_NDC1 perform best. However, when the propagation probability p is set to 0.05, the spreading scale of evolutionary processes is slightly different, as shown in Figure 3. In Figure 3, it can be seen that of all the experimental networks, the DPS_-NDC2 and DPSO_NDC3 have the best performance, while DPSO_NDC4 has the worst performance.

Another interesting thing is that we discover a saturation effect in improved local search strategy based on the degree centrality of neighbourhood nodes. That is, it is not that the large the hop range, the better the search effectiveness. More precisely, the best search performance of the algorithm is concentrated in the range of 3-hop, and the algorithm with the NDC area of 4-hop has the worst performance. It can be seen that the improvement of the DPSO algorithm with the local search strategy based on neighbourhood degree centrality is not that the larger the range, the better the performance.

All of the above-observed phenomena illustrate that the method of the local search approach is very critical for solving the global optimal solution in the DPSO algorithm. In addition, the largest spreading scale is the local search strategy with 3-hop area nodes degree centrality, 2-hop second and 4-hop is the worst performer. This phenomenon illustrates that there exists a saturation effect when considering seed nodes' neighborhood degree centrality. Considering the performance and computational cost of the algorithm, the DPSO_NDC algorithm adopts the neighbourhood degree centrality of the 3-hop area to improve the local search strategy.

D. COMPARISON OF TYPICAL ALGORITHMS

In order to evaluate the effectiveness of DPSO_NDC algorithm, the five state-of-the-art above-mentioned algorithms are used as baseline solution algorithms. After the above algorithms obtained the global optimal seed node-set, Mont Carlo(MC) simulation is used to evaluate the spreading performance of the algorithms under the IC model. The simulated evolutionary performances, as shown in Figure 4 and Figure 5. Figure 4 is the evolution curve when the propagation probability $p = 0.01$, and Figure 5 is the evolution curve when the propagation probability $p = 0.05$.

From the simulated curves, we can observe that the DPSO_NDC algorithm achieves comparable performance. It is worth mentioning that the proposed algorithm is more effective than the original DPSO algorithm. In Condmat and Pgp networks, the influence diffusion scale of the DPSO_NDC is even better than that of greedy algorithm, as shown in Figure 4(a) and (f), which illustrates the DPSO_NDC is rather effective for influential nodes

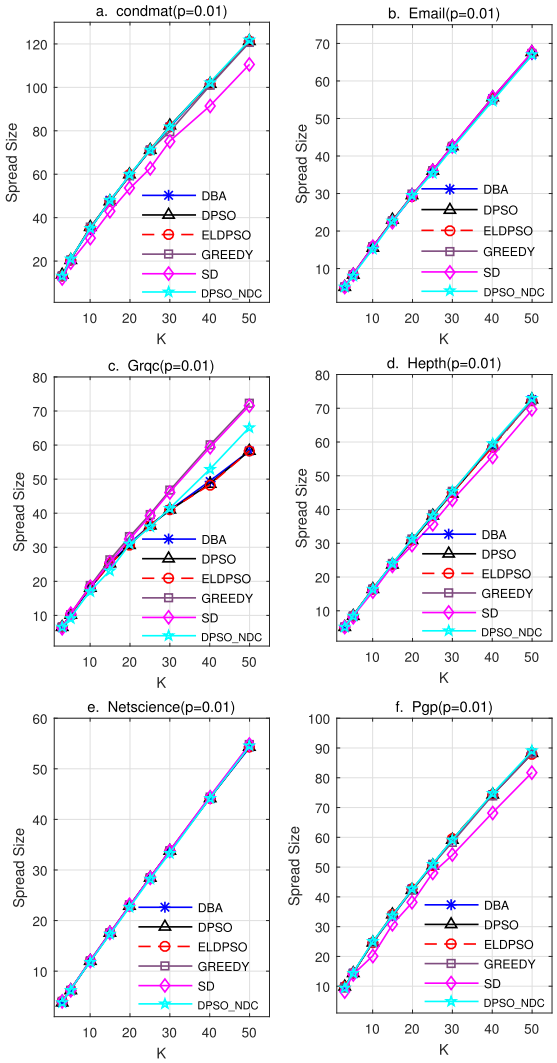

FIGURE 4. Line graph of influence diffusion scale in the six experimental social networks.(The horizontal axis K represents the scale of seed nodes, and the vertical axis represents the scale of nodes affected by the set of seed nodes identified by different algorithms in the MC simulation experiment when the propagation probability P=0.01.)

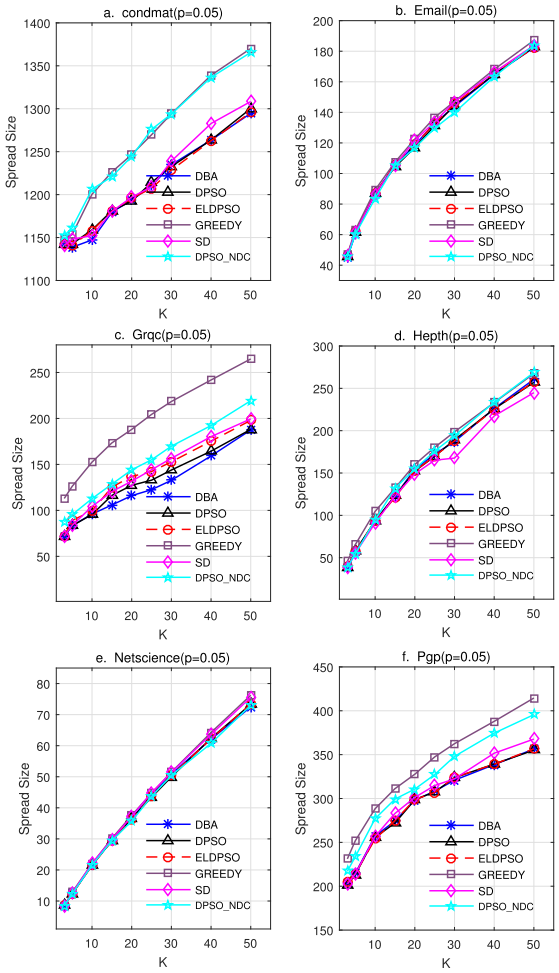

FIGURE 5. Line graph of influence diffusion scale in the six experimental social networks.(The horizontal axis K represents the scale of seed nodes, and the vertical axis represents the scale of nodes affected by the set of seed nodes identified by different algorithms in the MC simulation experiment when the propagation probability P=0.05.)

identification in the large-scale networks. In particular, the DPSO_NDC algorithm performs more effectively than the original DPSO and ELDSO algorithms in all six experimental networks. From the influence propagation scale obtained by simulating the seed node set, we can observe that the greedy algorithm has the most outstanding performance when $p = 0.05$, it is evident in Figure 5(c) and Figure 5(f). However, the computational complexity is the highest of all experimental algorithms. Again, the Algorithm DBA is not stable enough because of its random selection strategy, which is evident obviously in Figure 5(f). In summary, the algorithm DPSO_NDC is more effective in large-scale social networks and the size of seed node set is increasing.

E. WILCOXON RANK SUM TEST RESULTS

To detect the statistical difference between the art algorithm and the proposed algorithm, we performed the Wilcoxon rank-sum test with the significance level being 5% and 10%, respectively. The test uses the experimental values of seed node sets of different sizes under the IC model, The Wilcoxon sum-test results in different experimental networks between *DPSO_NDC* and other state-of-the-art algorithms are summarized in Table 3. In the test results in Table 3, the $p-value$ is a two-sided Wilcoxon rank-sum test, $h-value$ is a logical value indicating the test decision. If $h = 1$ then reject the null hypothesis and accept the alternative hypothesis. The result $h = 0$ indicates a failure to reject the null hypothesis at the specified significance level. For example, if p=0.058 in the test result, h=0 in the case of significance level α=0.05, indicating that there is no significant performance difference between the two algorithms. If the significance level α=0.1 then h=1, indicating that there is a significant difference in the performance of the two algorithms. From the statistical results in Table 3, we can observe that when the significance level is 10%, there is a significant difference between *DPSO_NDC* algorithm and all algorithms except

TABLE 3. Statistical results of the wilcoxon rank sum test for DPSO_NDC and other baseline algorithms at α=5% and α=10% significance level.

DPSO_NDC vs.	Test value type	Networks					
		NetScience	Email	HepTh	PGP	Grqc	CondMat
Greedy	$p-value$	0.7881	0.7892	0.7164	0.0779	0.0819	0.0642
	$h-value(\alpha$=0.1)	0	0	0	1	1	1
	$h-value(\alpha$=0.05)	0	0	0	0	0	0
DPSO	$p-value$	0.0962	0.0837	0.0739	0.0096	0.0323	0.0260
	$h-value(\alpha$=0.1)	1	1	1	1	1	1
	$h-value(\alpha$=0.05)	0	0	0	1	1	1
EDPSO	$p-value$	0.0611	0.0211	0.0056	0.0119	0.0455	0.0299
	$h-value(\alpha$=0.1)	0	1	1	1	1	1
	$h-value(\alpha$=0.05)	0	1	1	1	1	1
DBA	$p-value$	0.0091	0.0041	0.0073	0.0019	0.0064	0.0055
	$h-value(\alpha$=0.1)	1	1	1	1	1	1
	$h-value(\alpha$=0.05)	0	1	1	1	1	1
SD	$p-value$	0.0078	0.0086	0.0076	0.0086	0.0091	0.0032
	$h-value(\alpha$=0.1)	1	1	1	1	1	1
	$h-value(\alpha$=0.05)	1	1	1	1	1	1

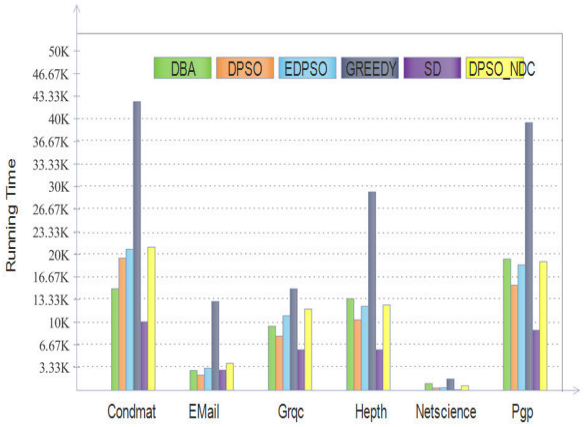

FIGURE 6. When the seed-node set size k=30 and the propagation probability p=0.01, the running time comparison diagram of the six algorithms on the six experimental social networks, respectively.

Greedy algorithm in all experimental networks. When the significance level is set to 5%, *DPSO_NDC* algorithm is significantly different from other experimental algorithms except for Greedy and DPSO algorithm.

F. COMPARISONS OF THE RUNNING TIME

In this section, the comparison of processing time (the case of 50 seed nodes) under the propagation probability $p = 0.01$ is conducted and as shown in Figure 6. We can observe that the Greedy algorithm is the most time-consuming. It takes 42K seconds, 13K seconds, 15K seconds, 29K seconds, 1.7K seconds and 39.5K seconds in the six experimental networks, respectively. Furthermore, as shown in Figure 6, we can observe that the computing time of DPSO_NDC is higher than that of the initial DPSO. This is because the DPSO_NDC algorithm takes considerable time to calculate the nearest neighborhood degree centrality of the 3-hop range, while initial DPSO algorithm only calculates the degree value of the temporary seed node itself. However, compared to the EDPSO algorithm, their running time is roughly the same. We remark that the running time of the proposed algorithm DPSO_NDC has a significant advantage over the greedy algorithm. For what concerns the running time, the SD algorithm has the best performance. However, the SD cannot provide the global optimal seed nodes. This is because the SD algorithm simply filters the influence of nodes from large to small without considering the overall influence of the top-k seed node-set.

V. CONCLUSION AND FUTURE WORK

Identifying top-k influential nodes in the social network remains a challenging task, especially as the network grows in size. It is another kind of effective exploration to solve such a problem by discretizing the appropriate swarm intelligence algorithm according to the structural characteristics of the network. The proposal of the DPSO algorithm is one of the important explorations. The original DPSO algorithm adopts the greedy mechanism of the local search strategy. This local search strategy can make the algorithm converge to the global optimal solution to a certain extent. However, this greedy local search strategy based on degree value does not consider the local network structure characteristics of candidate seed nodes, so it is very easy to fall into the global suboptimal solution. In order to make full use of the local structure of the network to enhance the local search ability of the algorithm, we propose a local search strategy based on the nearest neighbourhood degree centrality to avoid prematurity of the algorithm. We designed different algorithms based on the 0-hop to the 4-hop range of neighbourhood nodes to enhance the local search ability and conducted experiments in the six real social networks. The simulated results show that the neighbour node domain is most effective for the global optimal solution in the two-hop to the three-hop area. Whereas, the performance of the local search strategy based on the four-hop neighborhood degree centrality is significantly reduced. The above findings fully demonstrate that the nearest neighborhood degree centrality based on the 3-hop range has a significant effect on the performance improvement of the initial DPSO algorithm and 2-hop second. Meanwhile, the neighborhood degree centrality has a saturation effect in searching global optimal solution

of top-k influence nodes identification. The experimental results under the IC model illustrate that the performance of the proposed DPSO_NDC algorithm outperforms the original DPSO and ELDPSO, and its performance is better than the greedy-based algorithm in some scenarios.

There are several directions for future investigations. First, the literature of swarm intelligence algorithm is very rich, and each kind of swarm intelligence algorithm has its unique characteristics and advantages, such as chaos-based firefly algorithm (CFA) [45], crow search algorithm (CSA) [46], and so on. Therefore, in the following research, we will carefully study other related swarm intelligence algorithms to solve the problem of identifying the top-k influential nodes in the super-large-scale network. Furthermore, there are more realistic and complex influence propagation models than the IC model. Therefore, it is efficient and realistic to apply a discrete swarm intelligence algorithm to top-k influence node identification under other information propagation models.

REFERENCES

[1] B. Di Martino, K. C. Li, L. T. Yang, and A. Esposito, Eds., "Trends and strategic researches in Internet of everything," in *Internet of Everything. Internet of Things* (Technology, Communications and Computing). Singapore: Springer, 2018, doi: 10.1007/978-981-10-5861-5_1.

[2] C.-Y. Wu, M.-B. Hu, R. Jiang, and Q.-Y. Hao, "Effects of road network structure on the performance of urban traffic systems," *Phys. A, Stat. Mech. Appl.*, vol. 563, Feb. 2021, Art. no. 125361.

[3] L. Han, Q. Zhou, J. Zhang, X. Yang, R. Zhou, and J. Tang, "Polymorphism and consistency: Complex network based on execution trace of system calls in linux kernels," *Int. J. Mod. Phys. C*, vol. 31, no. 9, Sep. 2020, Art. no. 2050126.

[4] Y. Cho, J. Hwang, and D. Lee, "Identification of effective opinion leaders in the diffusion of technological innovation: A social network approach," *Technol. Forecasting Social Change*, vol. 79, no. 1, pp. 97–106, Jan. 2012.

[5] R. Bond, C. Fariss, J. Jones, A. Kramer, and C. Marlow, "A 61-million-person experiment in social influence and political mobilization," *Nature*, vol. 489, no. 7415, pp. 295–298, Sep. 2012.

[6] W. Wei, Z. Sun, H. Song, H. Wang, X. Fan, and X. Chen, "Energy balance-based steerable arguments coverage method in WSNs," *IEEE Access*, vol. 6, pp. 33766–33773, 2018, doi: 10.1109/ACCESS.2017.2682845.

[7] W. Wei, H. Song, H. Wang, and X. Fan, "Research and simulation of queue management algorithms in ad hoc network under DDoS attack," *IEEE Access*, vol. 5, pp. 27810–27817, 2017, doi: 10.1109/ACCESS.2017.2681684.

[8] A. I. E. Hosni, K. Li, and S. Ahmad, "Minimizing rumor influence in multiplex online social networks based on human individual and social behaviors," *Inf. Sci.*, vol. 512, pp. 1458–1480, Feb. 2020.

[9] S. Aral and D. Walker, "Identifying influential and susceptible members of social networks," *Science*, vol. 337, no. 6092, pp. 337–341, Jul. 2012.

[10] J. F. Robles, M. Chica, and O. Cordon, "Evolutionary multiobjective optimization to target social network influentials in viral marketing," *Expert Syst. Appl.*, vol. 147, Jun. 2020, Art. no. 113183.

[11] B. Zhao, C. Xu, S. Liu, J. Zhao, and L. Li, "Dynamic traffic bottlenecks identification based on congestion diffusion model by influence maximization in metro-city scales," *Concurrency Comput., Pract. Exper.*, vol. 9, Oct. 2020, Art. no. e5790.

[12] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Netw.*, vol. 1, no. 3, pp. 215–239, Jan. 1978.

[13] P.-L. Yang, G.-Q. Xu, Q. Yu, and J.-W. Guo, "An adaptive heuristic clustering algorithm for influence maximization in complex networks," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 30, no. 9, Sep. 2020, Art. no. 093106.

[14] C. Guo, L. Yang, X. Chen, D. Chen, H. Gao, and J. Ma, "Influential nodes identification in complex networks via information entropy," *Entropy*, vol. 22, no. 2, p. 242, Feb. 2020.

[15] Y. Wu, Y. Yang, F. Jiang, S. Jin, and J. Xu, "Coritivity-based influence maximization in social networks," *Phys. A, Stat. Mech. Appl.*, vol. 416, pp. 467–480, Dec. 2014.

[16] J. Zhao, Y. Song, and Y. Deng, "A novel model to identify the influential nodes: Evidence theory centrality," *IEEE Access*, vol. 8, pp. 46773–46780, 2020.

[17] G. Maji, A. Namtirtha, A. Dutta, and M. Curado Malta, "Influential spreaders identification in complex networks with improved K-shell hybrid method," *Expert Syst. Appl.*, vol. 144, Apr. 2020, Art. no. 113092.

[18] Q. Jiang, G. Song, C. Gao, Y. Wang, and W. Si, "Simulated annealing based influence maximization in social networks," in *Proc. 25th AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Aug. 2011, pp. 127–132.

[19] M. Gong, J. Yan, B. Shen, L. Ma, and Q. Cai, "Influence maximization in social networks based on discrete particle swarm optimization," *Inf. Sci.*, vols. 367–368, pp. 600–614, Nov. 2016.

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

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

[22] S. K. Kundu, C. A. Murthy, and S. K. Pal, "A new centrality measure for influence maximization in social networks," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.*, 2011, pp. 242–247.

[23] D. Chen, R. Xiao, and A. Zeng, "Path diversity improves the identification of influential spreaders," *Europhys. Lett.*, vol. 104, Jan. 2014, Art. no. 68006.

[24] J. Zhao, Y. Wang, and Y. Deng, "Identifying influential nodes in complex networks from global perspective," *Chaos, Solitons Fractals*, vol. 133, Apr. 2020, Art. no. 109637.

[25] F. Liu, Z. Wang, and Y. Deng, "GMM: A generalized mechanics model for identifying the importance of nodes in complex networks," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105464.

[26] T. Wen, D. Pelusi, and Y. Deng, "Vital spreaders identification in complex networks with multi-local dimension," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105717.

[27] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. Van Briesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 420–429.

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

[29] A. Salehi and B. Masoumi, "KATZ centrality with biogeography-based optimization for influence maximization problem," *J. Combinat. Optim.*, vol. 40, no. 1, pp. 205–226, Jul. 2020.

[30] A. Şimşek and R. Kara, "Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks," *Expert Syst. Appl.*, vol. 114, pp. 224–236, Dec. 2018.

[31] C. P. Sankar, A. S., and K. S. Kumar, "Learning from bees: An approach for influence maximization on viral campaigns," *PLoS ONE*, vol. 11, no. 12, Dec. 2016, Art. no. e0168125.

[32] J. Tang, R. Zhang, P. Wang, Z. Zhao, L. Fan, and X. Liu, "A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104833.

[33] A. Zareie, A. Sheikhahmadi, and M. Jalili, "Identification of influential users in social network using gray wolf optimization algorithm," *Expert Syst. Appl.*, vol. 142, Mar. 2020, Art. no. 112971.

[34] S. S. Singh, K. Singh, A. Kumar, and B. Biswas, "ACO-IM: Maximizing influence in social networks using ant colony optimization," *Soft Comput.*, vol. 24, no. 13, pp. 10181–10203, Jul. 2020.

[35] A. Sheikhahmadi and A. Zareie, "Identifying influential spreaders using multi-objective artificial bee colony optimization," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106436.

[36] J. Tang, R. Zhang, Y. Yao, Z. Zhao, P. Wang, H. Li, and J. Yuan, "Maximizing the spread of influence via the collective intelligence of discrete bat algorithm," *Knowl.-Based Syst.*, vol. 160, pp. 88–103, Nov. 2018, doi: 10.1016/j.knosys.2018.06.013.

[37] Y. Liu, M. Tang, T. Zhou, and Y. Do, "Identify influential spreaders in complex networks, the role of neighborhood," *Phys. A, Stat. Mech. Appl.*, vol. 452, pp. 289–298, Jun. 2016.

[38] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 2, 2006, Art. no. 036104.

[39] R. Guimer, L. Danon, and A. Dazguilera, "Self-similar community structure in a network of human interactions," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, Dec. 2003, Art. no. 065103.
[40] F. Lu, W. Zhang, L. Shao, X. Jiang, P. Xu, and H. Jin, "Scalable influence maximization under independent cascade model," *J. Netw. Comput. Appl.*, vol. 86, pp. 15–23, May 2017.
[41] S. Gregory, *Finding Overlapping Communities Using Disjoint Community Detection Algorithms*. Berlin, Germany: Springer, 2009.
[42] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, p. 2, Mar. 2007.
[43] D. Chakrabarti and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Comput. Surv.*, vol. 38, no. 1, p. 2, 2006.
[44] J. Tang, R. Zhang, Y. Yao, F. Yang, Z. Zhao, R. Hu, and Y. Yuan, "Identification of top-k influential nodes based on enhanced discrete particle swarm optimization for influence maximization," *Phys. A, Stat. Mech. Appl.*, vol. 513, pp. 477–496, Jan. 2019.
[45] I. Fister, M. Perc, S. M. Kamal, and I. Fister, "A review of chaos-based firefly algorithms: Perspectives and research challenges," *Appl. Math. Comput.*, vol. 252, pp. 155–165, Feb. 2015.
[46] A. Askarzadeh, "A novel Metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016, doi: 10.1016/j.compstruc.2016.03.001.

LIHONG HAN received the master's degree in computer software and theory from Lanzhou Jiaotong University, in 2009. He is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Lanzhou University. His research interests include social network analysis, machine learning, optimization algorithm, and real-time operating systems.

QINGGUO ZHOU received the B.S. and M.S. degrees in physics and the Ph.D. degree in theoretical physics from Lanzhou University, in 1996, 2001, and 2005, respectively. He is currently a Professor with Lanzhou University, where he is working with the School of Information Science and Engineering. His research interests include safety-critical systems, embedded systems, and real-time systems. He is also a Fellow of IET. He was a recipient of the IBM Real-Time Innovation Award in 2007, the Google Faculty Award in 2011, and the Google Faculty Research Award in 2012.

JIANXIN TANG received the Ph.D. degree in statistics from the School of Information Science and Engineering, Lanzhou University, in 2019. He is currently an Associate Professor with the Lanzhou University of Technology. His research interests include social network analysis, data mining, and machine learning.

XUHUI YANG received the Ph.D. degree from the School of Information Science and Engineering, Lanzhou University, in 2019. His research interests include machine learning, intelligent chip, parallel computing, the IoT, safety-critical systems, embedded systems, and real-time systems.

HENGJUN HUANG received the Ph.D. degree in statistics from the Capital University of Economics and Business, in 2013. He is currently a Professor with the Lanzhou University of Finance and Economics. His research interests include mathematical statistics, machine learning, and artificial neural networks.