

Identifying Influential Spreaders in Social Networks through Discrete Moth-Flame Optimization

Lu Wang, Lei Ma, Chao Wang, Neng-gang Xie, Jin Ming Koh, Kang Hao Cheong

Abstract—Influence maximization in a social network refers to the selection of node sets that support the fastest and broadest propagation of information under a chosen transmission model. The efficient identification of such influence-maximizing groups is an active area of research with diverse practical relevance. Greedy-based methods can provide solutions of reliable accuracy, but the computational cost of the required Monte Carlo simulations renders them infeasible for large networks. Meanwhile, although network structure-based centrality methods can be efficient, they typically achieve poor recognition accuracy. Here we establish an effective influence assessment model based both on the total valuation and variance in valuation of neighbor nodes, motivated by the possibility of unreliable communication channels. We then develop a discrete moth-flame optimization method to search for influence-maximizing node sets, using local crossover and mutation evolution scheme atop the canonical moth position updates. To accelerate convergence, a search area selection scheme derived from a degree-based heuristic is used. Experimental results on five real-world social networks, comparing our proposed method against several alternatives in current literature, indicates our approach to be effective and robust in tackling the influence maximization problem.

Index Terms—Social networks; influence maximization; assessment model; moth-flame optimization

I. INTRODUCTION

WITH the continued development of the Internet and Web 2.0 technologies, the modes of communication between people and the ways in which information is generated are rapidly evolving. In particular, social networking sites such as Facebook, Twitter, Instagram, and a growing landscape of microblogs have supported an increase in information consumption and sharing. The virtual world has, in effect, drastically shrunk distances in its physical counterpart.

C. Wang and K. H. Cheong are joint Corresponding Authors.

L. Wang, L. Ma, and N-g. Xie are with the Department of Management Science and Engineering, Anhui University of Technology, Anhui, Ma'anshan 243002, PR China.

C. Wang is with the Department of Architectural Engineering, Anhui Polytechnic University, Anhui, Wuhu 241000, PR China (email: wang.chao.cn.ah@gmail.com).

J. M. Koh and K. H. Cheong are with the Science, Mathematics and Technology Cluster, Singapore University of Technology and Design (SUTD), 8 Somapah Road, S487372, Singapore.

J. M. Koh is also with the California Institute of Technology, Pasadena, California 91125, United States.

K. H. Cheong is also with the SUTD-Massachusetts Institute of Technology International Design Centre, S487372, Singapore (email: kanghao_cheong@sutd.edu.sg).

Manuscript received TBC; revised TBC. This work was supported by the Scientific Research Foundation of Education Department of Anhui Province, China (Grant No. KJ2019A0091, KJ2019ZD09), Humanities and Social Science Fund of Ministry of Education of China (Grant No. 19YJAZH098) and the Singapore University of Technology and Design Start-up Research Grant (SRG SCI 2019 142). (Corresponding authors: Chao Wang; Kang Hao Cheong.)

Social influence drives human behavior offline and online and pervades the cultural market [1]. The proliferation of social networks has also provided enterprises with significant development opportunities in product promotion and sales [2]. Typically, these companies seek to market their products by selecting a group of initial users who influences their family and friends; if the users have sufficient connections to sustain an exponential spread of information, then the *viral marketing* campaign has succeeded [3]. Such social referrals play an important role in reshaping consumers' attitudes and behaviors [4]; indeed research by Chevalier *et al.* [5] shows that the reputation of customers can indeed directly affect the purchase behavior of consumers. To maximize profits, enterprises need to achieve maximum publicity at minimal expenditure. The identification of key nodes in social networks from which to launch campaigns is therefore important. This same problem arises in epidemic control, where the identification of super-spreaders is of interest, and is also relevant to the analysis of publications in science [6].

Influence maximization (IM) as treated here refers to the selection of an independent group of individuals, so that the number of individuals reachable through their social connections is maximized. There are, intuitively, numerous factors affecting information dissemination and communication, such as differing personal interests and socio-political alignment, but the characterization of these are typically unclear and subjective [7]. Thus, we generally approach the IM problem using only network structure information [8].

In the context of enterprise marketing, Domingos and Richardson had treated IM as an optimization problem, that can then be solved algorithmically [9]. A series of IM algorithms had then been successively proposed. Kempe *et al.* [10] adopted a combinatorial optimization approach and showed that the problem is generically NP-hard; other studies saw the development of greedy strategies in the independent cascade (IC) [11], [12] and linear threshold (LT) models [13] to solve problems of similar nature as IM, with a theoretical result that greedy methods can yield solutions with an accuracy not less than $1 - 1/e$ (63%) of the optimal. However, greedy methods require a large number of Monte Carlo information propagation simulations to be performed, which limits its applicability to large networks. In light of this, much effort had been dedicated to improving the efficiency of IM algorithms. For instance, Leskovec *et al.* [14] had developed an improved greedy-based strategy named the cost-effective lazy forward (CELF), which reduces computational cost by two orders of magnitude without detriment to accuracy by exploiting the sub-modularity of the objective function [15]. Subsequently,

Goyal *et al.* [16] proposed a further-optimized version named CELF++, with results indicating an efficiency improvement of $> 50\%$ in some cases over CELF.

Separately, heuristical methods may also be used to tackle IM problems. For instance, Chen *et al.* [17] had suggested a degree-discount heuristic based on the IC model and the degree centrality [18] metric. The essential observation is that in evaluating the influential power of a node, there is a tendency of over-estimation when there are overlaps between second-degree neighborhoods. A further development was made by Wang *et al.* [19] who proposed a stricter strategy, named *DegreePunishment*, to address the issue of neighborhood overlaps. A different shortest-path model (SPM) was introduced by Kimura *et al.* [20], who assessed the influential power of a node set on a basis of shortest-path lengths from the set to other nodes in the network, computed through Dijkstra's algorithm. Using a similar framework, Chen *et al.* [21] subsequently developed the maximum influence arborescence (MIA) model, which leverages local network structure to approximate information propagation and hence the influence of nodes. The model discards propagation paths with propagation probability below certain thresholds, and reduces computational cost to an extent. In a more recent work, Morone *et al.* [22] suggested that existing heuristical and structural centrality methods may neglect weakly connected nodes in complex networks, and hence proposed a collective optimization approach via mapping IM into an optimal percolation problem. Numerous works [23], [24] have indeed shown that network structure is an important consideration when evaluating influential power.

Much of recent research interest had been in the use of meta-heuristic algorithms, generically referring to constructing computationally-cheap influence evaluation models and treating the IM problem as one of fitness optimization, using appropriate optimization protocols. For example, Sankar *et al.* [25] had proposed an algorithm for influence maximization based on the *waggle dance* behavior of bees, and had verified the performance of the algorithm on a Twitter dataset. Cui *et al.* [26] had developed a degree-descending search strategy (DDSE) based on differential evolution and degree centrality metrics, exploring and identifying influential nodes through mutation, crossover, selection operators, and local search within neighborhoods; and Tang *et al.* [27] had introduced a discrete shuffling frog-leaping algorithm (DSFLA), which performs evolution through deterministic and random walks with a local degree-based replacement strategy. Gong *et al.* [28] had used local influence estimation (LIE) to approximate the influence of node sets in a two-hop neighborhood (the neighbors' neighbors), and applied discrete particle swarm optimization (DPSO) with degree centrality; simulation results indicate good performance, but the DPSO is susceptible to local minima entrapment. This is later addressed with an enhanced discrete particle swarm optimization (ELDPSO) algorithm, that exploits network topology around the seed nodes [29]. A discrete bat algorithm (DBA) based on DPSO's discrete coding criterion had also been explored [30] for promising results. The algorithm generates a candidate pool of nodes with potential influence according to the contribution

of nodes to the network topology to accelerate convergence.

The moth-flame optimization (MFO) algorithm is a new swarm intelligence optimization algorithm recently proposed by Mirjalili [31], which simulates the homing behaviour of moths near flame sources. The simplicity and efficiency of the algorithm prompted its application to problems in various fields; for example, Yousri *et al.* [32] utilized MFO to extract diode parameters in polycrystalline silicon solar modules, and Taher *et al.* [33] used an improved MFO to treat optimal power flow problems. There are numerous other benchmarking studies and applications of MFO, such as in the open shortest path first algorithm for routing in computer networks [34], in feature selection of machine learning [35], and to address loss minimization in optimal reactive power dispatch (ORPD) problems [36]. The results suggest that the MFO algorithm provides advantages in convergence speed and robustness.

The performance of MFO makes it an attractive choice for optimization in continuous search spaces; but there is also a wide range of optimization problems in discrete search space, such as flow routing and the traveling salesman problem, and indeed IM in social networks. There is currently little research into discrete versions of MFO, and no report on the application of MFO to IM problems. In light of this, we develop a discrete moth-flame optimization (DMFO) algorithm and apply it to IM, exploiting local network topology to efficiently identify node sets with maximal influence. The main contributions of the present paper are as follows—(1) a new solution representation for the IM problem; (2) the DMFO algorithm with automatic search area selection; (3) the proposed local crossover operator to enhance search space exploitation and optimization performance; (4) an influence estimation method based on both the total valuation and variance in valuation of nodes in a two-hop neighborhood, motivated by potential transmission unreliability across social connections, and which is used to construct a fitness function for optimization; and (5) experimental results comparing our approach to other available methods, across five real-world social network datasets. The results indicate that our approach is capable of an accuracy similar to that of greedy-based approaches, but at reduced computational cost. Compared to other IM methods based on meta-heuristics, our approach provides higher-quality results.

This paper is organized as follows. Section II gives a technical overview of the IM problem and existing methods, and Section III details our proposed approach. The experimental results and analyses are given in Section IV, followed by a conclusion and remarks on possible future work in Section V.

II. RELATED WORKS

A. Influence maximization

In the study of influence maximization (IM), a social network is abstracted as a graph $G = (V, E)$, where V is the set of all nodes in the network, and E is the edge set, representing the connections between nodes [37]. Given a positive integer $k < N = |V|$, one selects k nodes in the network as the seed set, with the aim of activating a maximum

number of nodes in the propagation process. In other words, the IM problem concerns the selection [9] of

$$\sigma(S^*) = \operatorname{argmax}(\sigma(S)), \quad (1)$$

where $\sigma(S)$ provides a measure of the influence of seed set S and $\sigma(S^*)$ is the influence-maximizing seed set, and $S^* \subseteq V$ and $|S^*| = k$.

B. Structural centrality

There are two primary questions concerning IM—(1) how to efficiently evaluate the influence of a node or set of nodes in a network, and (2) how to accurately find a group of nodes with maximal influence in a network. The literature previously discussed (Section I) has proposed a range of approaches to these problems. A family of methods concerns structural centrality, which facilitates the estimation of the influential power of a node or node set.

Degree centrality (DC) characterizes the direct influence of nodes [38], using the degree of the nodes as an evaluation metric. In essence, the assumption is that there is an association between the degree of a node in the network and its influence, since a larger number of social connections imply a greater ability for the node to spread information. The degree centrality of node i is

$$DC_i = \sum_{\substack{i \neq j \\ j \in V}} e_{ij}, \quad (2)$$

where e_{ij} is the adjacency between nodes i and j , that is, $e_{ij} = 1$ when i and j are connected and otherwise $e_{ij} = 0$.

Betweenness centrality (BC) [39] is an alternate measure of influence, and reflects the centrality of a node in a shortest-path sense. It posits that among the shortest paths of all node pairs in the network, the greater the number of shortest paths that pass through a node, the more important the node is—since it is more likely that the spread of information between any two parties in the network will pass through the node. Therefore betweenness centrality is a characterization of the control of a node over network flow along shortest paths [40]. The betweenness centrality of node i is defined as

$$BC_i = \sum_{\substack{i \neq m \neq n \neq i \\ m, n \in V}} \frac{L_{mn}^i}{L_{mn}}, \quad (3)$$

where L_{mn} is the number of all shortest paths from node m to n , and L_{mn}^i is the number of shortest paths from node m to n passing through node i .

Lastly, *closeness centrality* (CC) [41] uses the mean of the shortest distance from a node to any other in the network to evaluate its influence. It presupposes that the smaller the average shortest distance from a node to all others in the network, the more significant the node is, in essence characterizing the importance of a node through a measure of propagation time within the network. The closeness centrality of node i is

$$CC_i = \frac{N-1}{\sum_{\substack{i \neq j \\ j \in V}} d_{ij}}, \quad (4)$$

where d_{ij} is the shortest distance between nodes i and j .

Of relevance is also the PageRank algorithm [42], which evaluates website prominence based on random walks, and is used for website ranking in business scenarios. Like the centrality measures above, the influence of nodes in the network can be assessed by calculating their PR values. The PageRank formula for node i can be written

$$\begin{aligned} PR_i(T) &= s \sum_{j=1}^N \bar{a}_{ij} PR_j(T-1) + \frac{1-s}{N}, \\ \bar{a}_{ij} &= \begin{cases} \frac{1}{d_i^{\text{out}}} & \text{if } d_i^{\text{out}} > 0 \\ \frac{1}{N} & \text{if } d_i^{\text{out}} = 0, \end{cases} \end{aligned} \quad (5)$$

where s is a damping factor, T is the iteration number, and d_i^{out} is the out-degree of the node i in the network.

C. Meta-heuristic methods for IM

Generically, meta-heuristic IM algorithms seek to optimize a simplified fitness function in replacement of the computationally expensive Monte Carlo simulations typical in greedy strategies. Jiang *et al.* [43] had previously introduced an estimation model that sums the activation probability of all direct neighbors of a given seed set, termed the expected diffusion value (EDV), as an approximate measure of the influence of the seed set. The model is specified as

$$EDV(S) = k + \sum_{i \in N_S^{(1)} \setminus S} (1 - (1 - P)^{\tau(i)}), \quad (6)$$

where $N_S^{(1)} \setminus S$ is the set of direct neighbors of the seed set S excluding members of the seed set, and $\tau(i)$ is the number of repetitions of the neighbors of the node i and the members of S . When paired with a simulated annealing (SA) algorithm for optimization, EDV was shown to yield good solutions. Separately, the local influence estimator (LIE) as defined by Gong *et al.*—which is used in the ELDPSO algorithm—can be expressed as

$$LIE = k + \left(1 + \frac{\sum_{u \in N_S^{(2)} \setminus S} p_u^* d_u^*}{|N_S^{(1)} \setminus S|} \right) \sum_{i \in N_S^{(1)} \setminus S} \left(1 - \prod_{\substack{(i,j) \in E \\ j \in S}} (1 - p_{i,j}) \right), \quad (7)$$

where p_u^* is the IC-model activation probability, and d_u^* is the number of outgoing edges of node u within $N_S^{(1)}$ and $N_S^{(2)}$, where $N_S^{(2)}$ is the second-order neighborhood of the seed set S .

III. FITNESS FUNCTION AND ALGORITHM

Although all the technical methods reviewed above (Section II) have their own advantages, they do not address the issue of overlap between second-degree or higher neighborhoods (see Section I). We develop a more extensive estimation model to approximate the influence of a node set, serving as the fitness function to be optimized.

A. Fitness function

Pei *et al.* had observed that the influence of a node in a network depends largely on its information-spreading ability within its two-hop neighborhood, in that the propagation of information is mostly localized to within three degrees of separation [44], [45]. On this basis, we propose a method based on the total valuation and variance of valuation of activatable nodes in a two-hop neighborhood to estimate the informational influence of a seed set.

The activation probability of a node $i \in S^{1,2}$, with $S^{1,2}$ denoting the direct and second-order neighborhoods of a seed set S , is computed as

$$P(i) = P_{\text{neb}_1}(i) + P_{\text{neb}_2}(i), \quad (8)$$

where the first and second terms respectively calculate the probability that node i is activated by its direct and second-order neighbors in S . The first term is

$$P_{\text{neb}_1}(i) = 1 - (1 - p)^{\tau(i)}, \quad (9)$$

where p is the activation probability and $\tau(i)$ is the number of neighbors of node i that are in S . This functional form is adapted from the EDV model previously discussed. When not activated by its direct neighbors, node i can only be activated by its second-order neighbors. The probability that node i is activated by a direct neighbor $t \in S$ is

$$(1 - P_{\text{neb}_1}(i)) P_{\text{neb}_1}(t)p, \quad (10)$$

hence $P_{\text{neb}_2}(i)$ can be written

$$P_{\text{neb}_2}(i) = 1 - \prod_{\substack{t \in N_S^{(1)} \setminus S \\ e_{it} \in E}} [1 - (1 - P_{\text{neb}_1}(i)) P_{\text{neb}_1}(t)p]. \quad (11)$$

The valuation of node i and the total valuation of the seed set S are then

$$\begin{aligned} v(i) &= P(i)d_i, \\ W(S) &= \sum_{i \in S^{1,2}} v(i), \end{aligned} \quad (12)$$

where d_i is the degree of node i .

However, there is a subtlety in the present formulation. Since a large number of low-value nodes in the neighborhood and a small number of high-value nodes both contributes to maximizing $W(S)$ [8], using only $W(S)$ as the fitness function reflecting the influence of the seed set is not entirely satisfactory. Suppose there are two candidate seed sets in the same network, as illustrated in Fig. 1a (left) and 1b (left). Under the present IC-based model, the valuation of seed set $S_1 = \{1\}$ is $W(S_1) = (6 + 2 + 2 + 13)p = 6.9$, and that of seed set $S_2 = \{12\}$ is $W(S_2) = (3 + 3 + 3 + 12)p = 6.3$, with activation probability $p = 0.3$. Hence the influence of S_1 is deemed to be greater. However, in the propagation process, as shown in Fig. 1a (right), if node 10 fails to receive the message, then only nodes $\{2, 3, 4, 5\}$ are activated. In comparison, for seed set S_2 , even if a neighbor fails to activate, the overall influence of the seed set does not decline much, as shown in Fig. 1b (right). The reason for the decline in influence of S_1 is that the valuation distribution of its neighbor

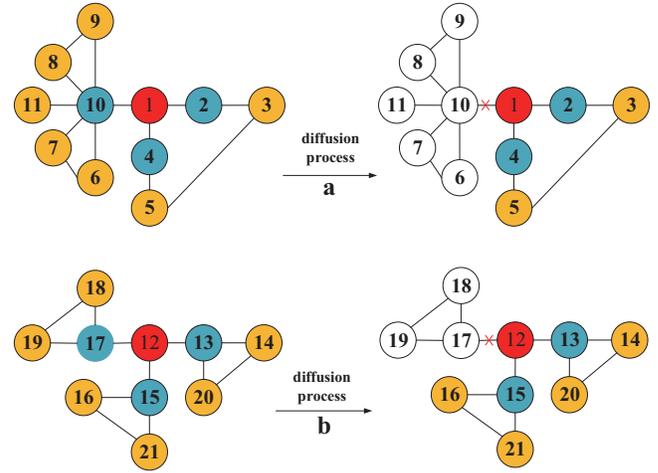


Fig. 1. Illustration of two candidate seed sets with similar fitness, but when the possibility of unreliable communication channels—that connections between individuals may stochastically fail—one seed set (in *a*) retains greater fitness and is therefore superior to the other (in *b*).

nodes is uneven, and this effect can be significant in large-scale networks containing greatly heterogeneous node degrees. To also consider the robustness of the seed set to transmission failure, we add the standard deviation of the valuation of activatable node to the fitness function. The complete fitness function is

$$F(S) = c_1 \cdot W(S) - c_2 \cdot \text{std}(v) \quad (13)$$

where c_1 and c_2 are weights, and $\text{std}(v)$ is the standard deviation of v , the valuation set of all activatable nodes.

With an increase in network size and therefore the number of activatable nodes, the magnitude of $W(S)$ will increase relative to $\text{std}(v)$, and the imposed penalty will cease to be effective. To counteract this, we construct the weights to be network size-dependent. For simplicity, we set $c_1 = 0.2$, and we have c_2 as

$$\begin{aligned} c_2 &= \begin{cases} 10^{F_2} & \text{for } N_1 < 4 \\ \omega \times 10^{F_2} & \text{for } N_1 \geq 4, \end{cases} \\ F_2 &= \begin{cases} 1 & \text{for } N_2 \leq 0 \\ N_2 & \text{for } 0 < N_2 < 3 \\ N_1 - 2 & \text{for } N_1 \geq 3, \end{cases} \\ N_1 &= \lfloor \log_{10} W(S) \rfloor, \quad N_2 = \lfloor \log_{10} \text{std}(v) \rfloor. \end{aligned} \quad (14)$$

Above, N_1 and N_2 respectively represent the number of digits of $W(S)$ and $\text{std}(v)$ minus one. Since N_1 scales with the size of the network and the number of activatable nodes, the penalty on $\text{std}(v)$ is retained across network scale. When the network size is small, the valuation and its variance between most nodes is small, which may cause the standard deviation N_2 to be negative. This reverses the effect of c_2 as a penalty weight. Hence we set the lower limit of the index F_2 to 1 so that $\text{std}(v)$ plays its intended role. Now, compared with $W(S)$, the $\text{std}(v)$ growth rate is relatively slow. When $N_1 \geq 3$, the numerical difference between N_1 and N_2 is obvious and will continue to grow. In this regime, is insufficient to only

Algorithm 1. DMFO algorithm framework

Input. Graph $G = (V, E)$, search area η , size of moth population n , size of seed set k , maximum number of iterations T_{\max} .

```

Initialize seed sets  $S$ 
Initialize moth position vectors  $M$ 
Initialize search area given  $\eta$ 
Initialize iterator  $T \leftarrow 0$ 
while  $T < T_{\max}$  do
  Compute fitness of members of seed sets  $S$  and sort
  Update moth position vectors  $M$ 
  Compute crossover locations and update seed sets to  $S'$ 
  Elite finalization of seed sets  $S : S_i \leftarrow \max(S_i, S'_i)$ 
  Apply mutation to seed sets
   $T \leftarrow T + 1$ 
end while
Output. Seed set with maximal influence found
    
```

rely on N_2 to adjust the magnitude of c_2 ; a simple solution is to use N_1 , which guarantees the penalty effect of $\text{std}(v)$ is retained in a larger-scale network. Lastly, ω provides a small amplification on the 10^{F_2} penalty weight in larger networks. Experiments on the effect of ω will be conducted in Section IV.

B. Optimization algorithm

The moth-flame optimization (MFO) framework is inspired by the homing flight behaviour of moths near fire sources. The moths are represented as point-particles in search space, with positions and velocities adjusted in a logarithmic spiral-like pattern to explore candidate solutions. In a recent work, Mirjalili [31] had proposed the update criteria for moth position M_i as follows,

$$\begin{aligned}
 D_i &= |F_j^T - M_i^T|, \\
 M_i^{T+1} &= D_i e^{\lambda \cos(2\pi\lambda)} + F_j^T,
 \end{aligned} \tag{15}$$

where D_i is the distance from the i^{th} moth to the j^{th} flame, F_j is the position of the j^{th} flame, T is the iteration number, and λ is a random number between $[-1, 1]$.

The original MFO runs on a continuous search space, but the search space in IM problems is discrete. We therefore propose a modification of MFO, accordingly named the discrete moth-flame optimization (DMFO) algorithm. In our implementation each moth i is associated with a candidate seed set S_i , and we utilize a local crossover and mutation scheme to evolve the seed sets over time. Algorithm 1 describes the overall framework of the DMFO algorithm.

1) *Population initialization:* In order to make full use of the topology of the network, we adopt a heuristical degree-based strategy to select the initial members of the seed set. To ensure diversity in the initial seed set, we multiply the degree of each node in the search area by a random factor $x \in (0, 1)$ and arrange them in descending order. Then, the top- k nodes are selected. This means the chance of each node being in the initial seed set is, on average, proportional to its degree; and there is also a chance for lower-degree nodes to be selected.

2) *Search area selection:* In general, selecting a reasonable search area is important in swarm intelligence algorithms; in the current IM context, the scale of the networks involved renders the trivial assignment of the entire graph as the search space infeasible. There is a trade-off involved in the

Algorithm 2. Seed set update rules

```

Input. Search area  $\eta$ , size of moth population  $n$ , seed set  $S$ , moth position vectors  $M$ 
for each  $i \leq n$  do
   $m_{ij} = \text{sig}(M_{ij})$ 
  Compute crossover location  $X$ 
  Perform cross-exchange on  $S_i$  to produce  $S'_i$ 
  while  $S'_i$  has duplicate element  $j$  do
     $S'_{ij} \leftarrow \text{replace}(v, \eta)$ 
  end while
end for
Sort seed sets according to fitness
for each  $i \leq n$  do
  if  $\text{fitness}(S'_i) > \text{fitness}(S_i)$  do
     $S_i \leftarrow S'_i$ 
  end if
end for
Output. Updated seed set  $S'$ .
    
```

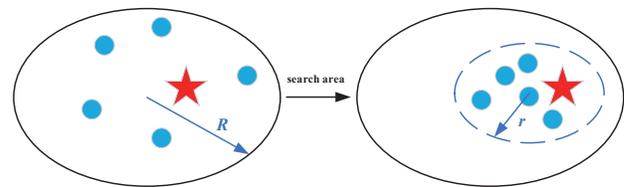


Fig. 2. High-level principle of the search area mechanism. The red star represents the optimal solution, the blue circles represent search agents.

selection—if the search area is too small then the algorithm converges quickly but easily gets trapped in a local optimum, and conversely if the area is too large then convergence becomes slow. Generically, a reasonable selection scheme reduces the size of the search domain whilst retaining most candidate optimal solutions (Fig. 2). To avoid the pitfalls at either extreme, we utilize a search area selection that scales with different seed set sizes. Nodes in the network are sorted in descending order of their degree and the arrangement is stored as D ; the top $\eta = kc$ nodes are then selected as the search area, where c is an amplification factor.

We initialize $c = 4$. Traditional greedy strategies are known to produce results of reliable accuracy, hence we use them to characterize the appropriate values of c ; this is only a one-off calibration procedure in the development of the algorithm, and does not incur computational cost when the algorithm is subsequently used. First, the greedy strategy (CELF++) is used to find the maximal-influence node set using different seed set sizes k , and the position of the node with the smallest degree in D is then taken as the upper bound. We present the resulting statistics on five social network datasets—NetScience, Email, Hamsterster, Ego-facebook, and Pages-public-figure [46], [47], [48], [48], [49]—in Table I.

The bold entries in Table I indicate that the upper bound positions for NetScience, Email and Ego-facebook networks rise sharply when k exceeds certain values. However, we found that the node degree distributions in these networks are in actuality reasonably clustered; the ill-behavior of the upper bounds can be attributed to the presence of a small number of outliers of much smaller degree than others. We

TABLE I
SEARCH AREA UPPER BOUND POSITIONS FOR INITIALIZATION PARAMETER CHARACTERIZATION

Networks/ k	3	6	9	12	15	18	21	24	27	30
NetScience [46], [47], [48]	3	11	28	28	28	69	313	313	323	323
NetScience (<i>o-rem.</i>)	3	11	28	28	28	69	313	313	313	313
Email [46], [47], [48]	6	8	28	28	31	142	142	142	142	142
Email (<i>o-rem.</i>)	6	8	28	28	31	72	72	72	72	72
Hamsterster [46], [47], [48]	4	8	10	15	44	44	61	61	61	61
Ego-facebook [49]	3	8	16	18	21	24	51	94	94	164
Ego-facebook (<i>o-rem.</i>)	3	8	16	18	21	24	51	94	94	94
Pages-public-figure [46], [47], [48]	6	10	10	14	17	19	37	42	42	69

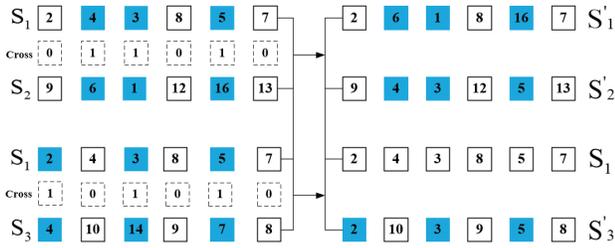


Fig. 3. Diagram illustrating the crossover operation. The blue boxes represent the nodes to be exchanged; the row labeled "cross" represents the crossover location vector, denoted X in equations.

hence refine the statistics to exclude these outliers, labelled (*o-rem.*). Clearly, $c = 4$ gives a good balance between providing a sufficiently large search space for solution diversity and maintaining reasonable computational cost.

3) *Seed set update rules:* To adapt the MFO algorithm to function on a discrete search space, we adopt a local crossover process to evolve the seed set from one iteration to the next, with the moth positions of MFO serving to determine the crossover location. The moth population is constructed with a random scatter in search space as

$$M = \text{rand}_{nk}(1) \times (u_b - l_b) + l_b, \quad (16)$$

where $\text{rand}_{nk}(1)$ is an $n \times k$ matrix of randomly generated numbers within $[0, 1)$, n is the moth population size, k is the size of the seed set, u_b and l_b respectively are imposed upper and lower bounds on entries of the moth position vectors. Note that after the update of moth positions M at each iteration, we use the standard sigmoid function, denoted $\text{sig}(\cdot)$, to normalize its entries [50]. The sigmoid map is continuous and strictly monotonic; saturation occurs when input $x \notin [-5, 5]$, hence we set $u_b = 5$ and $l_b = -5$ in Eq. (16) for simplicity.

From the position vector of the moths, the crossover location X is obtained as

$$X = \mathbf{C}(w_1 \cdot \mathbf{g}(m_1) + w_2 \cdot \mathbf{g}(m_{i+1})), \quad i = 1, 2, \dots, n-1, \quad (17)$$

$$\mathbf{g}(x) = \begin{cases} 0 & \text{if } x > \text{rand}(1) \\ 1 & \text{otherwise,} \end{cases}$$

$$\mathbf{C}(x) = \begin{cases} 0 & \text{if } x < 2 \\ 1 & \text{otherwise,} \end{cases}$$

where m_i is the position vector of moth i after mapping. For example, in a case of coefficients $w_1 = 1.4$ and $w_2 = 0.8$ and supposing $\mathbf{g}(m_i) = (0, 1, 1, 0, 1, 1)$ and $\mathbf{g}(m_{i+1}) = (0, 1, 1, 0, 1, 0)$, the crossover location is computed to be $X = (0, 1, 1, 0, 1, 0)$, which indicates to cross-exchange the second, third, and fifth elements of candidate seed set S_i with the current optimal (S_1). This local crossover process is illustrated in Fig. 3; note that while the current optimal (S_1) and best suboptimal (S_2) undergoes a two-way exchange, all other suboptimal seed sets are subject only to a one-way replacement from S_1 , to ensure that the best-known optimal does not degrade from iteration to iteration. In comparison with traditional crossover operators in genetic algorithms, the mostly one-way sharing of information here biases the update process to move towards the best-known optimal. The new seed sets are then sorted according to fitness, in the process selecting the new current optimal seed set. Algorithm 2 summarizes this update process, where $\text{replace}(v, \eta)$ indicates to randomly select a node v from η for replacement of duplicates.

4) *Mutation operation:* To allow the algorithm to explore a larger number of candidate nodes during the search process, we apply a mutation operation with small probability $Z = 0.1$ at the end of each iteration. This improves the ability of the algorithm to escape local minima entrapment. In the mutation operation, we randomly select q nodes from the first three candidate seed sets (S_1, S_2, S_3) and replace them with any node in η , ensuring that there are no duplicates in the seed set after replacement.

C. Computational complexity

As described, the DMFO algorithm comprises the steps of initialization, computation of fitness of candidates and sorting, location updates, and the application of the crossover-mutation evolution scheme. The initialization stage can be completed in $\mathcal{O}(nk)$ time, and the computation of fitness requires $\mathcal{O}(nk\bar{d}d^{(2)})$ time, where \bar{d} is the average degree and $d^{(2)}$ is the second-order degree of the node. Using quicksort, the fitness sorting can be done in $\mathcal{O}(n^2)$ time in the worst case, and updating the moth positions and determining the crossover locations take $\mathcal{O}(nk)$ time. In the crossover-mutation scheme, replacement operations are sometimes required to eliminate node duplication, and mutation is probabilistic with worst-case $\mathcal{O}(nk)$ complexity. Hence, overall, the time complexity of DMFO scales as $\mathcal{O}(n(k\bar{d}d^{(2)} + n)T_{\max})$.

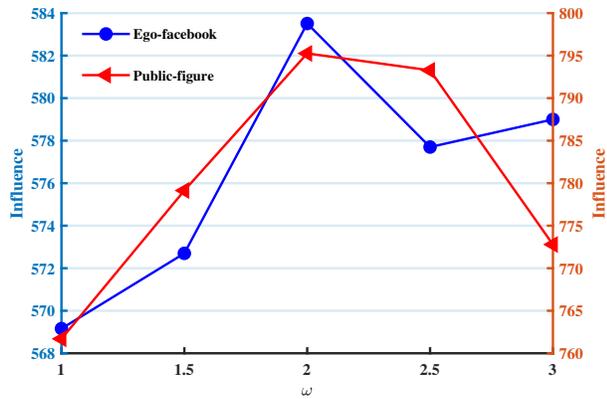


Fig. 4. Variations of the best-known maximal influence with different ω values.

IV. RESULTS AND ANALYSIS

To validate our proposed DMFO-based approach, we conducted experiments on five real-world network datasets and compared our results against several state-of-the-art methods, namely DC [38], BC [39], PageRank [42], simulated-annealing expected diffusion value (SAEDV) [43], Grey Wolf Optimization(GWO)[51], Degree-Descending Search Evolution(DDSE)[26], Enhanced Discrete Particle Swarm Optimization(ELDPSO)[29] and the greedy-based CELF++ [16]. The selected networks and their basic characteristics are listed in Table II, where C is the average clustering coefficient and p is the IC-model activation probability, set according to network size and average node degree.

TABLE II
ATTRIBUTES OF THE FIVE REAL-WORLD NETWORKS USED FOR VALIDATION EXPERIMENTS.

Networks	$ V $	$ E $	\bar{d}	C	p
NetScience	379	917	4.823	0.798	0.1
Email	1133	5451	9.622	0.254	0.05
Hamsterster	2426	16631	13.711	0.538	0.03
Ego-facebook	4039	88234	43.691	0.605	0.02
Pages-public-figure	11565	134152	11.600	0.179	0.04

A. Parameter configuration

We first investigate the effect of ω on the performance of our DMFO-based algorithm. To determine the best value of ω , we run the algorithm on two networks with $n = 30$, $k = 30$, $T_{\max} = 150$, $Z = 0.1$, while $\omega \in [1, 3]$. The maximal influence obtained under different ω are shown in Fig. 4. While there are noticeable variations in the influence-maximizing solutions obtained by the algorithm on the two networks as ω is varied, it is generically observed that the best results are achieved around $\omega = 2$. We therefore choose $\omega = 2$ as an appropriate value.

We now investigate the effect of the number of iterations T_{\max} and moth population size n on the performance of the algorithm. The obtained experimental results for the five networks are shown in Figs. 5 and 6.

Fig. 5 presents a comparison of the best-known maximal influence discovered during search and the running time of DMFO across different T_{\max} and a fixed $n = 10$. The influence-maximizing solutions improve slightly as T_{\max} is increased, but beyond $T_{\max} \gtrsim 150$ the marginal improvements may no longer justify the accompanying increase in computational cost. We therefore set $T_{\max} = 150$ as a compromise henceforth. In Fig. 6 we likewise observe diminishing returns as n is increased; in some networks, such as NetScience, Email, and Hamsterster, the improvement in maximal influence with increasing n is essentially too small to be noticeable. We therefore set $n = 30$ henceforth.

The parameters for the other algorithms were adapted from literature or otherwise determined through a preliminary comparative study. In particular, the damping factor of the PageRank algorithm was set as $s = 0.85$ and the number of iterations to 1000. In SAEDV, the initial temperature was set as $T_0 = 5$, termination temperature $T_f = 0.08$, number of inner cycles $q = 300$, and cooling rate $\Delta T = 0.02$. In ELDPSO, we set $n = 30$ and $T_{\max} = 100$, learning factors $c_1 = c_2 = 2$, and the inertia weight is set to 0.8. In DDSE, we set $n = 10$, $T_{\max} = 200$, and the probabilities of mutation, crossover and diversity operations are 0.1, 0.4 and 0.6 respectively. The number of Monte Carlo simulations for CELF++ was set to 10000; for all other algorithms, the influential power (Section IV-C) of the optimal seed sets were evaluated through 1000 Monte Carlo simulations. We explore seed set sizes $3 \leq k \leq 30$. To be fair in the comparison of algorithm performance and running time, all programs were written in MATLAB and run on a Windows 10 machine with a Core i7 3.2 GHz processor.

B. DMFO variants

To verify that the proposed search area selection (Section III-B2) indeed aids in finding optimal seed sets, we compared our DMFO algorithm against a variant without the selection scheme, named DMFO-na. The maximal fitness obtained on the five networks are presented in Table III.

On the NetScience network, it is observed that DMFO yields superior optimization results in comparison to DMFO-na across the entire range $3 \leq k \leq 30$, but the difference in performance is not large. Essentially, DMFO-na can still find reasonably good solutions in small networks. In the larger Hamsterster network, DMFO-na likewise finds reasonably good suboptimal solutions for $3 \leq k \leq 9$, and this is attributed to the degree-based initialization strategy, which provides a good initial approximation; as k increases, the performance gap between DMFO and DMFO-na becomes more obvious. In the Ego-facebook and Pages-public-figure networks, the fitness results suggest DMFO-na to be superior to DMFO in some cases, namely when $3 \leq k \leq 9$ and $k = 6$ respectively; but this is not the full picture. The issue lies in the relaxation of the valuation variance constraint previously discussed (Section III-A), arising from a circumvention of weights in the fitness function. Since DMFO-na has no restrictions in search area, some nodes with low valuation near the edges of the network may be randomly selected to become seed set members,

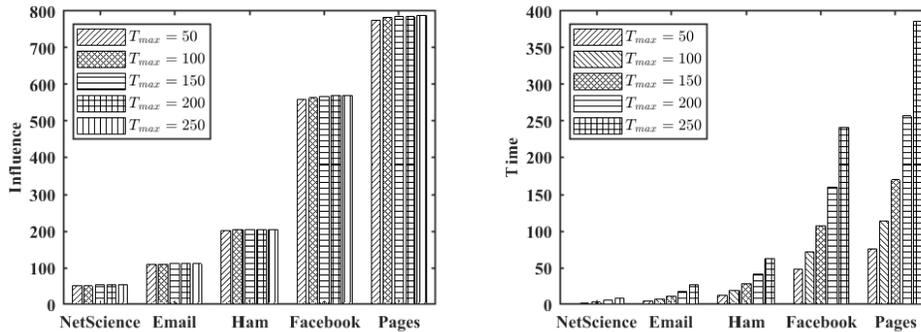


Fig. 5. Comparison of best-known maximal influence (left) and running time of our DMFO-based algorithm (right) on the five real-world network datasets for different iteration numbers T_{max} .

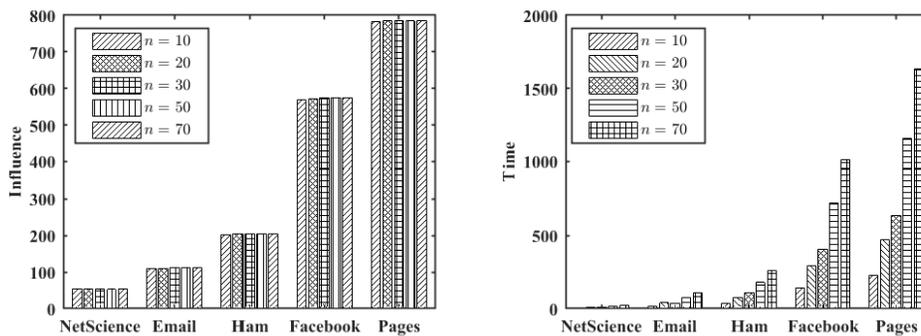


Fig. 6. Comparison of best-known maximal influence (left) and running time of our DMFO-based algorithm (right) on the five real-world network datasets for different moth population sizes n .

resulting in a diminishing value of N_1 and a tipping of the desired balance between the magnitude and the variance of valuations. This allows access to seed sets of higher numerical fitness, but which are undesirable. In comparison, DMFO searches only in the space of nodes with large degree, so N_1 better accounts for network size. In general, it can be seen that the search area selection scheme proposed here does indeed aid in the algorithm performance, especially as seed set size k is increased.

C. Performance comparison

To demonstrate the performance of our DMFO algorithm in IM, we compare optimization results against eight other state-of-the-art methods from literature under the IC model (see Fig. 7). We use a common metric of *influentiality*, evaluated as the total number of activated nodes divided by the network size, to assess performance across the different algorithms. It is seen that the node sets identified through DMFO achieve good influential power across all five networks. Furthermore, the optimal results found by DMFO is smooth without significant fluctuations as k is varied, suggesting good robustness across use cases.

In the NetScience network of Fig. 7a, the node set from DMFO is of similar influentiality as that from CELF++. The SAEDV and DDSE algorithm is competitive with DMFO and CELF++ at higher k ranges, but at lower k its performance

declines; with the exception of ELDPSO, the remaining four methods yield noticeably inferior results across the entire k range. Note when $k \gtrsim 27$, the results from DC and SAEDV appears to stagnate. This is attributable to the limitations of DC and EDV—the former considers only node degree in the evaluation of influential power, and the latter considers only first-order neighbor activation probabilities, hence at large k the problem of overlapping second-degree neighborhoods (Section I) becomes significant, to the detriment of solution quality.

The nine algorithms yield similar results for the Email network dataset of Fig. 7b. The DC and PageRank algorithms fare slightly worse than SAEDV, DDSE and ELDPSO; BC and GWO performs worst across the entire k range. Our DMFO algorithm, CELF++ and DDSE yield the best results. Though the difference is small, we note DMFO surpasses CELF++ when $k \gtrsim 12$. This indicates that DMFO can effectively tackle IM problems when the activation probability (p) of the network is small. In the Hamsterster network of Fig. 7c, DMFO, CELF++ , SAEDV and DDSE all achieve comparable results. When $k \lesssim 24$, DMFO and CELF++ are both outperformed by SAEDV, DDSE and ELDPSO, and for larger k the accuracy of DC diminishes to below PageRank. Note GWO performed noticeably worse than all other algorithms at any given seed size in the networks.

Lastly, Figs. 7d and 7e present the performance of the algo-

TABLE III
COMPARISON OF BEST-KNOWN FITNESS FUNCTION VALUES ACHIEVED BY DFMO AND DMFO-NA ALGORITHMS

Network	Methods	k									
		3	6	9	12	15	18	21	24	27	30
NetScience	DMFO	7.37	16.68	22.67	26.83	30.37	33.03	35.55	36.95	38.97	39.85
	DMFO-na	7.28	16.32	21.05	23.06	26.90	28.31	33.64	34.39	36.68	37.65
Email	DMFO	55.77	97.79	130.34	157.28	179.08	197.48	216.33	228.39	244.08	256.06
	DMFO-na	55.39	95.89	127.65	150.13	172.11	191.06	205.91	217.52	233.16	246.35
Hamsterster	DMFO	295.00	456.50	575.16	670.47	747.11	812.43	864.06	909.83	947.25	977.14
	DMFO-na	295.00	455.87	574.16	657.97	728.83	782.56	840.03	887.06	926.55	965.55
Ego-facebook	DMFO	678.04	1209.31	1636.29	1923.94	2170.60	2355.02	2550.60	2658.64	2741.66	2992.54
	DMFO-na	1812.72	1842.78	1818.41	1855.38	2076.07	2208.46	2323.42	2392.58	2511.95	2579.85
Pages-public-figure	DMFO	1668.07	666.43	1104.25	1443.93	1747.07	1983.84	2210.50	2386.87	2528.52	2671.63
	DMFO-na	1667.97	1825.57	1024.03	1369.71	1693.07	1864.22	2080.17	2202.29	2384.39	2472.35

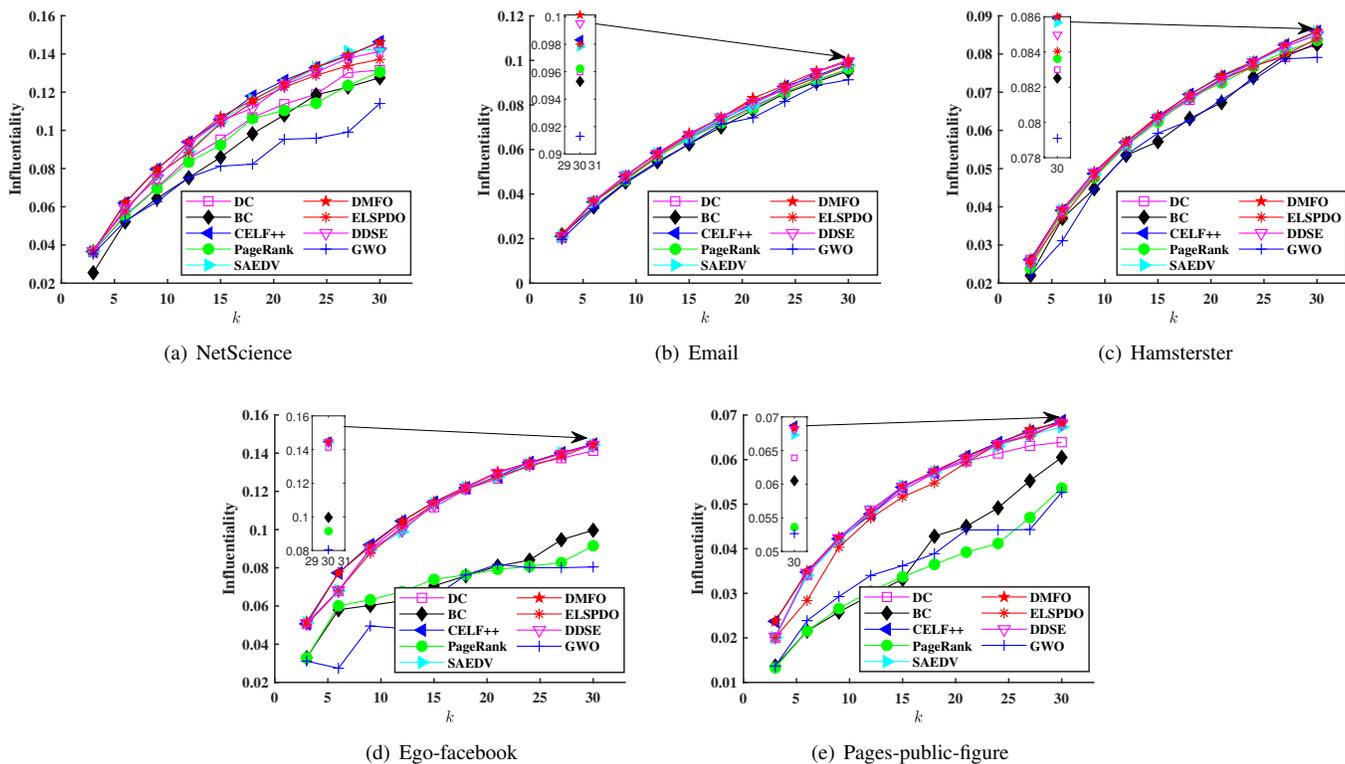


Fig. 7. Performance comparison of our DMFO-based approach against eight other algorithms in current literature, across five real-world social network datasets under the IC model. The metric used in assessing node set optimality is informativity, the number of activated nodes divided by network size. Above k is the size of the seed set.

gorithms on the Ego-facebook and Pages-public-figure networks, which are larger in size (see Table II). It is observed that BC and PageRank perform poorly across the board. On Ego-facebook, DMFO and CELF++ provide similar optimization results; SAEDV, ELSPDO and DDSE also becomes competitive at larger k values, but is not as good when $k \lesssim 15$. On Pages-public-figure, which is the largest network amongst the five, DMFO is once again on-par or superior to CELF++, and both are better than SAEDV and DC. When $k \gtrsim 18$, the relative stagnation in the results of DC can be attributed to its inherent limitation in neglecting neighborhood overlaps, as discussed previously. Notably, GWO performed poorly on both networks; in some cases the maximal influence found decreased with increasing k . This may be caused by fitness

mis-estimation and changes in network topology—the fitness function of GWO only maximizes the number of activatable nodes, and is not tightly integrated with the diffusion model. It hence cannot effectively treat situations with small activation probabilities in the IC model. Overall, the results obtained here strongly suggests excellent performance by our DMFO-based approach, that is robust across different network sizes, and node set size k .

D. Running time comparison

To further assess the performance of our approach, we now compare the running times of the nine algorithms across the five real-world networks (Fig. 8), setting $k = 30$. Since DC needs only to sort nodes in the network by their degrees and

then select the top- k nodes as the seed set, its computational cost is expected to be the lowest, as is indeed observed. Although the greedy-based CELF++ is highly accurate, it requires many (tens of thousands) of Monte Carlo simulations to reliably evaluate marginal improvements in seed sets, hence it is considerably computationally expensive. The BC and PageRank algorithms are fast for small networks, but their cost scales rapidly with network size, and the results they yield were clearly inferior to DMFO. Importantly, our DMFO-based approach achieves on-par or superior node set identifications as CELF++, but is computationally cheaper; in effect we achieve the same quality of IM solutions more efficiently.

in convergence, it also introduces a risk of neglecting weakly-connected nodes in the network during optimization search. Striking a balance between the advantages and disadvantages of restricting the search area and design a more reasonable search area may be of significance in further improving the algorithm. Secondly, in practice, there may be a need to optimize multiple objectives simultaneously. Therefore, extending DMFO to multi-objective optimization in the IM context can be a focus of future research. Finally, there are many other NP-hard discrete optimization problems—applying the DMFO-based approach to other domains may prove fruitful. This may concern, for example, the refinement of feature sets for data mining tasks, the optimization of locations and paths in logistics distribution, reliability-constrained minimization of operational costs in industries, and the optimization of tool and spray paths in manufacturing. Such cross-domain extensions from the current work can be considered in the future.

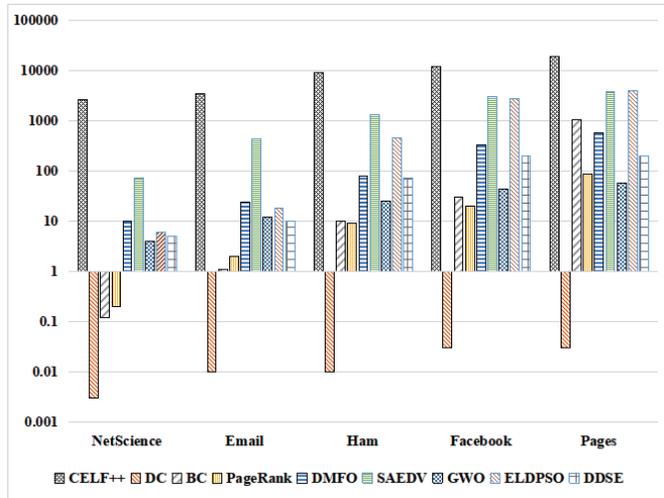


Fig. 8. Comparison of algorithm running time, measured in seconds, for the nine methods examined in this study, across the five network datasets.

E. Statistical tests

To rigorously illustrate the advantages of our DMFO-based approach, we verify statistical significance in the difference in performance of the nine methods. We perform the Friedman [52], [53] test to determine statistical significance, and use the Holm [54] method as a post-hoc procedure to correct for multiple comparisons. The confidence level $\alpha = 0.05$ in all cases. The Holm procedure adjusts α values in a step-down manner—for each value of k , the N_a algorithms are sorted in ascending order according to their p -values; if the i^{th} p -value in the ordered list is below $\alpha/(N_a - i)$, the corresponding hypothesis is rejected and we compare the next p values. The test proceeds until a null hypothesis cannot be rejected, and all remaining hypotheses are then retained. Tab. IV reports the statistical analysis results, taking DMFO as the control method. Based on our analysis, the highlighted values indicate that there is no significant difference between the two compared methods. In all cases, DMFO provides similar performance to CELF++; this strongly supports our discussion in the previous section.

F. Future work

There are a few areas for potential future investigation. First, although the degree heuristic-based search area selection aids

V. CONCLUSIONS

In this paper, we have proposed a new hybrid MFO-based optimizer, called DMFO, for tackling influence maximization (IM) in social networks. We redefine the position vector updates of the MFO algorithm, and adopt a local crossover and mutation procedure to evolve seed set members from iteration to iteration, with crossover locations determined from moth positions. In addition, we also provide a search area selection scheme adaptive to seed set size, to improve the convergence speed of the algorithm. A new influence estimation model based on the total valuation and variance in valuation of neighbor nodes is designed to efficiently evaluate the influential power of seed sets, taking into consideration the possibility of transmission unreliability in social links. To preserve the desired constraint relation across different networks, the weights in the fitness function are adaptively adjusted according to network size. Experiments on five real-world social networks reveal that DMFO not only has good robustness and IM accuracy comparable to the state-of-the-art greedy strategy (CELF++), but also has a lower running time.

REFERENCES

- [1] J. Onnela and F. Reedssochas, “Spontaneous emergence of social influence in online systems,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 43, pp. 18 375–18 380, 2010.
- [2] E. Abrahamson and L. Rosenkopf, “Social network effects on the extent of innovation diffusion: A computer simulation,” *Organization Science*, vol. 8, no. 3, pp. 289–309, 1997.
- [3] J. Goldenberg, B. Libai, and E. Muller, “Talk of the network: A complex systems look at the underlying process of word-of-mouth,” *Marketing Letters*, vol. 12, no. 3, pp. 211–223, 2001.
- [4] J. J. Brown and P. H. Reingen, “Social ties and word-of-mouth referral behavior,” *Journal of Consumer Research*, vol. 14, no. 3, pp. 350–362, 1987.
- [5] J. A. Chevalier and D. Mayzlin, “The effect of word of mouth on sales: Online book reviews,” *Journal of Marketing Research*, vol. 43, no. 3, pp. 345–354, 2006.
- [6] L. Lu, D. Chen, X. Ren, Q. Zhang, Y. Zhang, and T. Zhou, “Vital nodes identification in complex networks,” *Physics Reports*, vol. 650, no. 650, pp. 1–63, 2016.
- [7] J. L. Payne and M. J. Eppstein, “Evolutionary dynamics on scale-free interaction networks,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 895–912, 2009.

TABLE IV
STATISTICAL RESULTS (p -VALUES) OF HOLM PROCEDURE FOR THE NINE COMPARED ALGORITHMS AT $\alpha = 0.05$.

Methods	k									
	3	6	9	12	15	18	21	24	27	30
DC	0.140	0.105	0.250	0.250	0.040	0.105	0.045	0.075	0.048	0.040
BC	0.042	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.014	0.007
GWO	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
CELF++	0.429	0.488	0.488	1.000	0.299	1.000	0.564	0.817	1.000	0.908
SAEDV	0.664	0.150	0.318	0.744	0.332	1.000	0.532	0.498	1.000	0.796
DMFO	-	-	-	-	-	-	-	-	-	-
DDSE	0.897	0.408	0.858	0.712	0.318	0.897	0.496	0.496	1.000	0.712
PageRank	0.048	0.018	0.006	0.024	0.000	0.018	0.000	0.000	0.036	0.018
ELDPSO	0.897	0.112	0.112	0.816	0.318	0.424	0.532	0.112	0.260	0.897

[8] A. Zareie, A. Sheikahmadi, and M. Jalili, "Identification of influential users in social network using gray wolf optimization algorithm," *Expert Systems With Applications*, vol. 142, p. 112971, 2020.

[9] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 61–70.

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

[11] D. Chen, L. Lu, M. Shang, Y. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," *Physica A-statistical Mechanics and Its Applications*, vol. 391, no. 4, pp. 1777–1787, 2012.

[12] J. Goldenberg, B. Libai, and E. Muller, "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*, vol. 9, no. 3, pp. 1–18, 2001.

[13] M. Granovetter, "Threshold models of collective behavior," *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[14] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, pp. 420–429.

[15] C. Qian, J. C. Shi, K. Tang, and Z. H. Zhou, "Constrained monotonek-submodular function maximization using multiobjective evolutionary algorithms with theoretical guarantee," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 99, pp. 595–608, 2017.

[16] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th international conference companion on World wide web*, 2011, pp. 47–48.

[17] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 199–208.

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

[19] X. Wang, Y. Su, C. Zhao, and D. Yi, "Effective identification of multiple influential spreaders by degreepunishment," *Physica A-statistical Mechanics and Its Applications*, vol. 461, pp. 238–247, 2016.

[20] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks," in *European conference on principles of data mining and knowledge discovery*. Springer, 2006, pp. 259–271.

[21] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 1029–1038.

[22] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, 2015.

[23] C. P. Galstyan A, Musoyan V, "Maimizing influence propagation in networks with company structure," *Physical Review E*, vol. 79, no. 5, pp. 056–102, 2009.

[24] J. Shang, S. Zhou, X. Li, L. Liu, and H. Wu, "Cofim: A community-based framework for influence maximization on large-scale networks," *Knowledge Based Systems*, vol. 117, pp. 88–100, 2017.

[25] C. P. Sankar, S. Asharaf, and K. S. Kumar, "Learning from bees: An approach for influence maximization on viral campaigns," *Plos One*, vol. 11, no. 12, p. e0168125, 2016.

[26] L. Cui, H. Hu, S. Yu, Q. Yan, Z. Ming, Z. Wen, and N. Lu, "Ddse: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks," *Journal of Network and Computer Applications*, vol. 103, pp. 119–130, 2018.

[27] 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," *Knowledge Based Systems*, vol. 187, p. 104833, 2020.

[28] M. Gong, J. Yan, B. Shen, L. Ma, and Q. Cai, "Influence maximization in social networks based on discrete particle swarm optimization," *Information Sciences*, vol. 367, pp. 600–614, 2016.

[29] 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," *Physica A-statistical Mechanics and Its Applications*, vol. 513, pp. 477–496, 2019.

[30] 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," *Knowledge Based Systems*, vol. 160, pp. 88–103, 2018.

[31] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge Based Systems*, vol. 89, pp. 228–249, 2015.

[32] D. Allam, D. Yousri, and M. B. Eteiba, "Parameters extraction of the three diode model for the multi-crystalline solar cell/module using moth-flame optimization algorithm," *Energy Conversion and Management*, vol. 123, pp. 535–548, 2016.

[33] M. Taher, S. Kamel, F. Jurado, and M. Ebeed, "An improved moth-flame optimization algorithm for solving optimal power flow problem," *International Transactions on Electrical Energy Systems*, vol. 29, no. 3, 2019.

[34] P. Garg and A. Gupta, "Optimized open shortest path first algorithm based on moth flame optimization," *Indian journal of science and technology*, vol. 9, no. 48, 2017.

[35] E. Emary and H. M. Zawbaa, "Impact of chaos functions on modern swarm optimizers," *PLOS ONE*, vol. 11, no. 7, 2016.

[36] M. H. Sulaiman, Z. Mustaffa, O. Aliman, H. Daniyal, and M. R. Mohamed, "Application of moth-flame optimization algorithm for solving optimal reactive power dispatch problem," *IET Conference Proceedings*, p. 16, January 2016.

[37] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 418–430, 2012.

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

[39] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," *Theory of Computing*, vol. 11, no. 1, pp. 105–147, 2015.

[40] Y. Liu, X. Wang, and J. Kurths, "Framework of evolutionary algorithm for investigation of influential nodes in complex networks," *IEEE Transactions on Evolutionary Computation*, 2019.

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

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

- [43] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie, "Simulated annealing based influence maximization in social networks," in *Proceedings of the twenty-fifth AAAI conference on artificial intelligence*, 2011, pp. 127–132.
- [44] S. Pei, L. Muchnik, J. S. Andrade, Z. Zheng, and H. A. Makse, "Searching for superspreaders of information in real-world social media," *Scientific Reports*, vol. 4, no. 1, pp. 5547–5547, 2015.
- [45] C. N.A. Christakis, J.H. Fowler, "the surprising power of our social networks and how they shape our lives," *Little, Brown*, 2009.
- [46] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 4292–4293.
- [47] R. A. Rossi and N. K. Ahmed, "Networkrepository: A graph data repository with visual interactive analytics," in *Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, USA*, 2015, pp. 25–30.
- [48] R. Rossi and N. Ahmed, "Network repository: A scientific network data repository with interactive visualization and mining tools," 2012. [Online]. Available: <http://networkrepository.com/>
- [49] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," <http://snap.stanford.edu/data>, Jun. 2014.
- [50] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *International Workshop on Artificial Neural Networks*. Springer, 1995, pp. 195–201.
- [51] A. Simsek and R. Kara, "Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks," pp. 224–236, 2018.
- [52] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Publications of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.
- [53] ———, "A comparison of alternative tests of significance for the problem of m rankings," *Annals of Mathematical Stats*, vol. 11, no. 1, pp. 86–92, 1940.
- [54] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.