



# A survey on meta-heuristic algorithms for the influence maximization problem in the social networks

Zahra Aghaee<sup>1</sup>  · Mohammad Mahdi Ghasemi<sup>2</sup> · Hamid Ahmadi Beni<sup>3</sup> · Asgarali Bouyer<sup>3</sup> · Afsaneh Fatemi<sup>1</sup>

Received: 16 September 2020 / Accepted: 24 March 2021 / Published online: 15 September 2021  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Austria, part of Springer Nature 2021

## Abstract

The different communications of users in social networks play a key role in effect to each other. The effect is important when they can achieve their goals through different communications. Studying the effect of specific users on other users has been modeled on the influence maximization problem on social networks. To solve this problem, different algorithms have been proposed that each of which has attempted to improve the influence spread and running time than other algorithms. Due to the lack of a review of the meta-heuristic algorithms for the influence maximization problem so far, in this paper, we first perform a comprehensive categorize of the presented algorithms for this problem. Then according to the efficient results and significant progress of the meta-heuristic algorithms over the last few years, we describe the comparison, advantages, and disadvantages of these algorithms.

---

✉ Zahra Aghaee  
Z.Aghaee@eng.ui.ac.ir

Mohammad Mahdi Ghasemi  
mohammadmehdi.ghasemi@gmail.com

Hamid Ahmadi Beni  
h.ahmadi@azaruniv.ac.ir

Asgarali Bouyer  
a.bouyer@azaruniv.ac.ir

Afsaneh Fatemi  
a\_fatemi@eng.ui.ac.ir

- <sup>1</sup> Department of Software Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran
- <sup>2</sup> Department of Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran
- <sup>3</sup> Department of Information Technology and Communications Azarbaijan, Shahid Madani University, Tabriz, Iran

**Keywords** Social networks · Influence maximization problem · Viral marketing · Influence spread · Meta-heuristic algorithm

**Mathematics Subject Classification** 91D30

## 1 Introduction

Recently, social networks have become increasingly popular among users. With the use of the increasing of the social network users, the effect of these networks on society has increased. Examples related to these effects are: preventing the spread of a particular disease, selling factory products, and advertising presidential candidates. In these examples, the essential problem is how to selecting influential users so that these users could maximize their influence on other users in the social network. This problem to Influence Maximization Problem (IMP) has modeled, and its goal is the finding of top-k influential nodes with maximum effect in a social network [1–4].

The IMP is one of the hard optimization problems first presented by Domingos and Richardson to identify influential customers that lead to more sales in viral marketing through word-of-mouth [5, 6]. In this marketing, with the goal of extensive advertising, the specific customers with the most influential in a short time among other individuals are selected [7–10]. The influence spread indicates the number of active individuals at the end of the diffusion process, and the running time indicates the time of selecting influential individuals during the diffusion process.

Kempe et al. [1] in the year 2003, proved that the IMP is an NP-hard problem and have presented a greedy algorithm to solve it named General Greedy algorithm. This algorithm guarantees an approximation factor  $1 - 1/e \simeq 0.63123$  [11]. Due to the high running time of the General Greedy algorithm, especially on large-scale graphs, the High Degree algorithm presented, which has a higher speed than the General Greedy algorithm [1], while its influence spread also is lower than the General Greedy algorithm. Then, the researchers presented different algorithms based on two essential challenges in the IMP: the influence spread and running time.

There is some survey like [12–17] on social influence analysis. The authors of [12] focus on the simulation-based, the proxy-based, and the sketch-based approaches. Banerejee et al. provided a comprehensive understanding of the IMP, target set selection in a social network and social influence [13]. Sun et al. presented a survey about statistical measurements related to social similarity and influence such as centrality [14]. Li et al. [15] investigated the research methods about epidemics and influence models such as the SI model, independent cascade model, linear threshold model. Pei et al. described some of the most important theoretical models and spreading processes [16]. Pastor-Satorras et al. provided a coherent presentation of epidemic processes [17]. These surveys are rather incomplete because an abundance of meta-heuristic algorithms have been developed. Also, the meta-heuristic algorithms are a trade-off between influence spread and efficiency. That's why compared with the existing surveys, this paper focuses on presenting a comprehensive survey on the meta-heuristic algorithms for the IMP.

These algorithms based on their methods and frameworks divided into five different categories. One of these is the meta-heuristic algorithms that have recently, with efficient results and significant progress in finding influential nodes, attracted the attention of researchers. Given the importance of these algorithms, in this paper, we first analyze the IMP algorithms, then, in particular, review the meta-heuristic algorithms for the IMP in social networks, and in the following, compare these algorithms with each other. Also, evaluations show that greedy algorithms are inefficient and the influence-path based algorithms improve the running time of greedy algorithms, but the running time of this algorithm needs further improvement in large-scale networks. Heuristic algorithms are very fast but do not have optimal penetration spread. Community-based algorithms to improve running time ignore inappropriate communities. The meta-heuristic algorithms are a trade-off between influence spread and efficiency.

The main purposes of this paper are:

- Providing a comprehensive categorize based on the IMP algorithms in social networks.
- Analysis of the IMP algorithms in social networks based on their methods.
- Reviewing and comparing different meta-heuristic algorithms for the IMP in social networks.

It is noteworthy that so far, no comprehensive comparison of the meta-heuristic algorithms for the IMP has been performed.

This paper is organized as follows: Sect. 2 describes the background of the IMP in the social networks. In Sect. 3, the IMP algorithms, and in particular, the meta-heuristic algorithms for the IMP, are reviewed. Sections 4 and 5 describes the discussion and general conclusions of the meta-heuristic algorithms for the IMP in social networks.

## 2 Background

In this section, we introduce the following two aspects: problem definition and different diffusion models.

### 2.1 Problem definition

Kempe et al. proved the IMP in social networks is an NP-hard problem and formulated it [1]. Hence, to find influential nodes according to the social network graph  $G(V, E)$ , the set  $S$  consists of  $k$  nodes is selected ( $S \subset V(G)$ ). The feature of these nodes is the information diffusion in the graph under a diffusion model to the largest number of nodes in the least time, where  $V(G) = \{v_1, v_2, \dots, v_n\}$  represents the set of graph nodes and  $E(G) = \{e_1, e_2, \dots, e_m\}$  represents the set of the graph edges. Also,  $n = |V(G)|$  is the total number of graph nodes and  $m = |E(G)|$  is the total number of graph edges. Also,  $k$  named seed node in set  $S$  and the size of the set  $S$  is  $|S| = k$ . So, the optimal seed set,  $S^*$ , is obtained based on Eq. (1) [1].

$$S^* = \operatorname{argmax}_{S \subset V, |S|=k} \sigma(S) \quad (1)$$

In Eq. 1,  $\sigma(S)$  is an objective function that gives the expected influence spread on the graph  $G(V, E)$  in the final of the diffusion process and  $S^*$  represents the  $k$  nodes with the maximum influence spread in the graph. Kempe et al. proved that the objective function  $\sigma(S)$  is monotonicity and submodular [1]. Thus, the function  $\sigma(S)$  is monotonicity, if Eq. (2) for any set  $S$  and any node  $v \notin S$  is established [12].

$$f(S \cup \{v\}) \geq f(S) \quad (2)$$

Also, the function  $\sigma(S)$  is submodular, if Eq. (3) is established when  $S \subseteq T$  [12].

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T) \quad (3)$$

## 2.2 Diffusion models

Information diffusion in social networks means spreading the data, rumor, knowledge, advertise, and the like that are done by the society individuals through their communication [18]. This process is one of the important problems in the analysis of the social network. As mentioned in the problem definition section, The IMP has done under a diffusion model. Researchers proposed different diffusion models in the social networks including Independent Cascade (IC) Model [1], Weighted Cascade (WC) Model [1], Linear Threshold (LT) Model [1], Susceptible-Infected (SI) model [19], Susceptible-Infected-Susceptible (SIS) model [19] and Susceptible-Infected-Recovered (SIR) model [19]. Of course, three diffusion models IC, WC, and LT are the most widely used models in the IMP. First, Kempe et al. used these three diffusion models for the IMP in social networks [1]. These three models are probabilistic diffusion models [10]. In the following, these diffusion models are described.

### 2.2.1 Independent Cascade (IC) Model

This model is used to model information cascades in social networks so that a node can only influence on nodes that it is connected to. Each node has two states: active or inactive. A node, after activated, can activate its neighbor nodes with activation probability  $p_{v,u} \in [0, 1]$ . Also, nodes change from inactive to active, but not vice versa. In other words, In the IC model, the node  $v$  that becomes active at the time  $t$  has, in the next time  $t + 1$ , one chance of activating each of its neighbor node  $u$  with activation probability  $p_{v,u}$ . The activated node generates a random number between zero and one for each its neighbor. The diffusion process continues until no other new nodes can be activated [7]. Algorithm 1, shows the pseudo-code related to the process of the IC model [19].

**Algorithm 1:** Independent Cascade (IC) model**Require:** graph  $G(V, E)$ , set of seed nodes  $A_0$ , activation probabilities  $p_{v,u}$ **Return** Final set of activated nodes  $A_\infty$ 


---

```

1:  $i = 0$ ;
2: while  $A_i \neq \{\}$  do
3:    $i = i + 1$ ;
4:    $A_i = \{\}$ 
5:   for all  $v \in A_{i-1}$  do
6:     for all  $u$  neighbor of  $v, u \notin \bigcup_{j=0}^i A_j$  do
7:        $rand = generate\ a\ random\ number\ in\ [0,1]$ ;
8:       if  $rand < p_{v,u}$  then
9:         activate  $u$ ;
10:         $A_i = A_i \cup \{u\}$ 
11:       end if
12:     end for
13:   end for
14: end while
15:  $A_\infty = \bigcup_{j=0}^i A_j$ 
16: Return  $A_\infty$ ;

```

---

### 2.2.2 Weighted Cascade (WC) Model

Different between the IC model and the WC model is their activation probability so that these two diffusion models work just the same as each other. The WC model has a changeable activation probability  $p(v, u) = 1/d_u$ , where  $d_u$  is the degree of node  $u$  in undirected graphs and is the in-degree of node  $u$  in directed graphs while activation probability in the IC model is the constant value [20].

### 2.2.3 Linear Threshold (LT) Model

In this diffusion model, each node  $v$  has a random threshold value ( $\theta_v \in [0, 1]$ ,  $v \in V$ ) and this threshold is independent of all other thresholds. Also, every  $u \in N(v)$  has a nonnegative weight  $w_{u,v} \leq 1$  that  $\sum_{u \in N(v)} w_{u,v} \leq 1$  that  $N(v)$  is the neighbours of node  $v$  ( $w_{u,v}$  is the influence weight on the edge  $(u, v)$ ). In each time  $t$ , the inactive node will become active, if  $\sum_{u \in N^a(v)} w_{u,v} \geq \theta_v$ , where  $N^a(v)$  is the set of active neighbours of node  $v$  [12]. Algorithm 2, shows the pseudo-code related to the process of the LT model.

**Algorithm2:** Linear Threshold (LT) model**Require:** graph  $G(V, E)$ , set of seed nodes  $A_0$ **Return** Final set of activated nodes  $A_\infty$ 


---

```

1:  $i = 0$ ;
2: while  $A_i \neq \{\}$  do
3:    $i = i + 1$ ;
4:    $A_i = \{\}$ 
5:   for all  $v \notin A_{i-1}$  do
6:     for all  $u$  neighbor of  $v, u \in \cup_{j=0}^i A_j$  do
7:        $\theta_v = \text{generate a random number in } [0,1]$ ;
8:       if  $\sum_{u \in N^a(v)} w_{u,v} \geq \theta_v$  then //  $N^a(v)$  is the set of active neighbours of node  $v$ 
9:         activate  $v$ ;
10:         $A_i = A_i \cup \{v\}$ 
11:       end if
12:     end for
13:   end for
14: end while
15:  $A_\infty = \cup_{j=0}^i A_j$ 
16: Return  $A_\infty$ ;

```

---

### 2.2.4 Susceptible-infected (SI), Susceptible-infected-susceptible (SIS) and susceptible-infected-recovered (SIR) models

These models are epidemic diffusion models. Some researchers have used the epidemic models to simulation the infection and recovery processes of nodes for the IMP in social networks. Disease diffusion starts from a set of infected seed nodes. The simplest epidemic model is the SI model. In this diffusion model, each node has two states: susceptible and infected. When a node has a susceptible state, it can potentially get infected by the disease, and when a node  $v$  is infected, it will remain infected forever and spreads disease to its susceptible adjacent node  $u$  with a probability  $w_{vu}$ . The SIS model is also similar to the SI model. Different between the SIS model, and the SI mode that is an infected node  $v$  can become susceptible again with a probability  $\gamma_v$ . Also, in the SIR model, each node has three states: susceptible, infected, and recovered. The SIR model is also similar to the SI model, but in the SIR model, when a node  $v$  is infected, it has a probability of  $\gamma_v$  to recover and becomes immune to the disease, which means  $v$  will not get infected any more [15].

## 3 The IMP algorithms in social networks

In this section, we introduce the following three aspects: categories of IMP Algorithms, theoretical analysis of the IMP algorithms, and a survey of meta-heuristic algorithms for the IMP in the social networks.

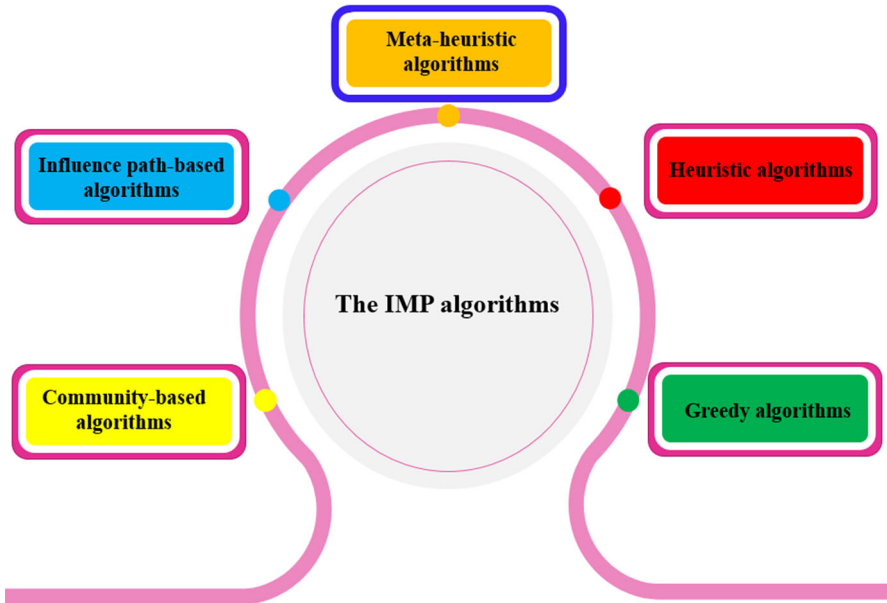


Fig. 1 The categorization diagram of the IMP algorithms in the social networks

### 3.1 Categorization of the IMP Algorithms in social networks

Researchers have presented different algorithms for finding influential nodes, that they have tried to improve the before presented algorithms. Researchers, different categorizations for these algorithms have been proposed [10, 12, 13, 21, 22]. In this survey, these algorithms based on their framework and method divided into five categories: (1) Greedy algorithms, (2) Heuristic algorithms, (3) Community-based algorithms, (4) Influence path-based algorithm and (5) Meta-heuristic algorithms. Figure 1, shows the categorization diagram of the IMP algorithms in social networks.

Greedy algorithms are often used to solve hard optimization problems. These algorithms are based on General Greedy algorithms that have been proposed to improve the inefficiency of General Greedy algorithms. The main idea in these algorithms is to use Monte Carlo simulation for influence spread calculations. The optimal selection of seed nodes depends on the number of repetitions of Monte Carlo simulation that in each round of Monte Carlo simulation, one node is added into the seed set that has the largest marginal contribution to Eq. (1). These algorithms make the best possible choice according to the problem conditions. Of course, their goal is to optimize the problem by continuing this approach. In many cases, a greedy process offers approximations of acceptable results with high accuracy and guarantees an optimal approximation factor, but these algorithms often have high running time, especially on the large-scale graphs. General Greedy [1], CELF [23], NewGreedyIC [24], NewGreedyWC [24], CELF + + [25], StaticGreedy [26] and SMG [27] algorithms are examples of greedy algorithms for the IMP. A theoretical analysis of these algorithms is shown in Table 1.

In this table, the general information of the IMP algorithms, including the category of each algorithm, the publication year of the algorithms, information diffusion model used, the time complexity of each algorithm, used data set, the scale of the data sets, and comparable algorithms is shown. The scale of the data sets are small, medium and large. Small, medium, and large scales are considered S, M, and L respectively. Also, ✓ means used, and ✗ means not used.

The heuristic algorithms can find good and near-optimal solutions at a low running time for hard optimization problems. Influential nodes have certain topological features in social networks, so the main idea in these algorithms is to use the graph topology feature on the selection of seed nodes. Algorithms [1, 18, 24, 28–30] and [31] use only one graph topology feature, while Algorithms [32, 33] and [6] use a combination of several graph topology features to select seed nodes. These algorithms are often used to solve problems to reduce the running time of greedy algorithms. High Degree [1], DD [24], PageRank [31], VoteRank [28], LIR [30], HybridRank [32], LGIM [33], IMMRA [6], and GIN [36] algorithms are examples of heuristic algorithms for the IMP. Table 2 shows a theoretical analysis of these algorithms. Researchers to improve the scalability in social network graphs, community-based algorithms for a finding of the influential nodes have presented. These algorithms, according to the similarities between the nodes of each community, such as having a common interest with each other, have been proposed. The main idea of a community-based algorithm is to reduce the computational overhead of influence spread by eliminating communities that are not suitable for selection of seed nodes. Also, in these algorithms, candidate nodes in each community are selected and then seed nodes are selected from the set of candidate nodes. CGA [37], IM-LPA [38], CoFIM [22], PHG [39], C2IM [40], ICIM-GREEDY[41], ComBIM [42] and TI-SC [43] algorithms are examples of community-based algorithms for the IMP. A theoretical analysis of these algorithms is shown in Table 3. Influence spread on some specific paths in the graph can increase. Therefore, researchers have presented different algorithms by identifying of the specific paths from the graph with the goal of increasing speed to the finding of the influential nodes and achieving high influence spread. The SPIM [44], LDAG [45], PMIA [46], SIMPATH [47], MIA [48] and IPA [49] algorithms are examples of influence path-based algorithms for the IMP. The main idea in these algorithms is to calculate the influence spread based on the diffusion path. Also, influence is limited to local regions, so these algorithms examine the diffusion path locally to reduce the computational overhead of influence spread in social networks. A theoretical analysis of these algorithms is shown in Table 4.

One of the main problems of heuristic algorithms is their fast convergence and being fall into local optimum points, so the researchers have presented meta-heuristic algorithms to solve the heuristic algorithms problem. The meta-heuristic algorithms provide acceptable and efficient results by having local optimum solutions and applications in various problems. The algorithms of SA [50], SAEDV [50], GNA [51], DPSSO [52], DDSE [53], DBA [54], ELDPSSO [55], ELDPSSO – CC [55], ELDPSSO – BC [55], ADPSSO [56], ADABC [57], GWIM [58] and DSFLA [59] are examples of meta-heuristic algorithms for the IMP. The main idea in meta-heuristic methods is:

- use the cost function with less computational complexity



**Table 1** The comparison the greedy algorithms for the IMP in the social networks

Category	Algorithm name	Year	Diffusion model							Time complexity	Data set name	Data set scale			Comparable algorithm
			IC	WC	LT	SI	SIS	SIR	S			M	L		
Greedy algorithms	General Greedy	2003	✓	✓	✓	✗	✗	✗	$O(kNRM)$	NetPHY	✗	✓	✗	High Degree and Random	
	CELF	2007	✓	✗	✗	✗	✗	$O(kNRM)$	Blog	✗	✗	✓	General Greedy		
	NewGreedyIC	2009	✓	✗	✗	✗	✗	$O(kRM)$	NetHEPT and NetPHY	✗	✓	✗	CELF, SD, DD, Random, High Degree, MixedGreedyIC, and Distance		
	NewGreedyWC	2009	✗	✓	✗	✗	✗	$O(kRTM)$	NetHEPT and NetPHY	✗	✓	✗	CELF, SD, DD, Random, High Degree, MixedGreedyWC and Distance		
	CELF++	2011	✓	✓	✗	✗	✗	$O(kNRM)$	NetHEPT and NetPHY	✗	✓	✗	CELF		
	StaticGreedy	2013	✓	✓	✗	✗	✗	$O(R'M + kNR'M')$	NetHEPT, NetPHY, DBLP, Epinions, Slashdot, and Douban	✗	✓	✓	CELF, PMIA, SPIM, DD, and High Degree		
	SMG	2014	✓	✗	✗	✗	✗	-	NetGR-QC and Wiki-Vote	✓	✓	✗	CELF		

**Table 2** The comparison the heuristic algorithms for the IMP in social networks

Category	Algorithm name	Year	Diffusion model							Time complexity	Data set name	Data set scale			Comparable algorithm
			IC	WC	LT	SI	SIS	SIR	S			M	L		
Heuristic algorithms	High Degree	2003	✓	✓	✓	✓	✓	✓	$O(kN)$	NetPHY	✓	✓	✓	Random and General Greedy	
	DD	2009	✓	✓	✓	✓	✓	✓	$O(k \log N + M)$	NetHEPT and NetPHY	✓	✓	✓	CELF, SD, MixedGreedyIC, Random, High Degree, NewGreedyIC and Distance	
	PageRank	2012	✓	✓	✓	✓	✓	✓	-	NetHEPT and NetPHY	✓	✓	✓	General Greedy, MixedGreedy, and NewGreedy	
	VoteRank	2016	✓	✓	✓	✓	✓	✓	-	YOUTUBE, COND-MAT, BERKSTAN and NOTRE DAME	✓	✓	✓	High Degree, ClusterRank, KshellRank, $C_2$ , H-index, PageRank, and LeaderRank	
	LIR	2017	✓	✓	✓	✓	✓	✓	$O(M + kN')$	NetGR-QC, Ca-HepTh, GP2P, and Power	✓	✓	✓	High Degree, Distance, NewGreedy, MixedGreedy, and DD	
	HybridRank	2018	✓	✓	✓	✓	✓	✓	$O( N  +  M )$	COND-MAT, DBLP, Epinions and Wiki-vote	✓	✓	✓	K-shell, Eigenvector, and High Degree	

Table 2 continued

Category	Algorithm name	Year	Diffusion model							Time complexity			Data set name	Data set scale			Comparable algorithm
			IC			WC			LT	SI	SIS	SIR		S	M	L	
			✓	✗	✗	✗	✗	✗									
LGIM		2019	✓	✗	✗	✗	✗	✗	✗	✗	$O(pop * M \log N)$	NetGR-QC, Wiki-Vote, NetHEPT, CA-CindMat, p2p-Gnutella31, and Epinions1	✓	✓	✓	DD, PMIA, IMM, and DDSE	
IMMRA		2020	✓	✗	✗	✗	✗	✗	✗	✗	-	DBLP and Facebook	✓	✗	✗	General Greedy	
GIN		2020	✓	✗	✗	✗	✗	✗	✗	✗	-	Email, Netscience, Power, NetHEPT, Net-PHY, Epinions1, and Slashdot0902	✓	✓	✓	General Greedy, High Degree, SAEDV, LJR, and DDSE	

**Table 3** The comparison the community-based algorithms for the IMP in social networks

Category	Algorithm name	Year	Diffusion model						Time complexity			Data set name	Data set scale			Comparable algorithm
			IC WC LT SI SIS SIR						IC	WC	MD		S	M	L	
			IC	WC	LT	SI	SIS	SIR								
Community-based algorithms	CGA	2010	✓	✗	✗	✗	✗	✗	✗	✗	-	Mobile	✗	✗	✓	MixedGreedy, NewGreedy, DD, and Random
	IM-LPA	2016	✓	✗	✓	✗	✗	✗	✗	✗	$O(T'MD)$	Football, SFI, Email, Facebook, and PGP	✓	✓	✗	General Greedy, High Degree, Betweenness, Closeness, Local centrality, K-shell and PageRank
	CoFIM	2017	✗	✓	✗	✗	✗	✗	✗	✗	$O(k^2 \cdot N \cdot k_{max})$	NetHEPT, NetPHY, Epinions, Amazon, DBLP, Patent, Pokec, LiveJournal, and Orkut	✗	✓	✓	IPA, TIM+, IMM, SD, and High Degree
PHG		2019	✗	✗	✓	✗	✗	✗	✗	$O(N \log N + N + kn'm')$	Amazon0302, Epinions, Brightkite, and Ca-HepTh	✓	✓	✓	HPG, General Greedy, PageRank, High Degree, and Random	

Table 3 continued

Category	Algorithm name	Year	Diffusion model					Time complexity	Data set name	Data set scale			Comparable algorithm
			IC	WC	LT	SI	SIS			SIR	S	M	
C2IM		2019	✓	✗	✓	✗	✗	✗	Dolphin, BHOSLIB, email-EU-core, Gnutella08, Astro-ph, AS-22JULY06, and Gnutella30	✓	✓	✗	Random, High Degree, CIM and TIM
CombIM		2019	✓	✓	✗	✗	✗	✗	Ca-HepTh, COND-MAT, and Epinions	✓	✓	✓	Random, High Degree, MAX_CLUS, DAG1-SPBP, and BSSA
TI-SC		2020	✓	✗	✗	✗	✗	✗	DBLP, Amazon, Ego-Facebook, and Email	✓	✗	✓	LIR, CI, DD, k-core, High Degree, and PHG

**Table 4** The comparison the influence-path based algorithms for the IMP in social networks

Category	Algorithm name	Year	Diffusion model						Time complexity	Data set name	Data set scale			Comparable algorithm
			IC	WC	LT	SI	SIS	SIR			S	M	L	
Influence path-based algorithms	SPIM	2006	✓	✗	✗	✗	✗	✗	$O(kNM)$	BN and CN	✗	✓	✗	High Degree, PageRank, Closeness, and Betweenness
	LDAG	2010	✗	✗	✓	✗	✗	✗	$O(Nt_{\theta} + kt_{\theta}m_{\theta}(m_{\theta} + \log N))$	NetHEPT, Epinions, Amazon, and DBLP	✗	✓	✓	General Greedy, SPIN, DD, and PageRank
	PMIA	2010	✗	✓	✗	✗	✗	✗	-	NetHEPT, DBLP, Epinions, and Amazon	✗	✓	✓	General Greedy, DD, SPIM, PageRank and Random
	SIMPATH	2011	✗	✗	✓	✗	✗	✗	$O(kNP_{\theta})$	NetHEPT, Last.fm, Flixster and DBLP	✗	✓	✓	High Degree, PageRank, CELF, LDAG and CELF +
	MIA	2012	✗	✓	✗	✗	✗	✗	-	NetHEPT, DBLP, Epinions, and Amazon	✗	✓	✓	General Greedy, DD, SPIM, PageRank, and Random
	IPA	2013	✓	✓	✗	✗	✗	✗	$O\left(\frac{NO_1n_{uu}}{c'} + k^2\left(\frac{O_1n_{uu}}{c'} + (c' - 1)\right)\right)$	Epinion, Stanford, DBLP, Patent, and LiveJournal	✗	✗	✓	CELF, PMIA, SD, and Random

- use of candidate nodes, including highly centralized nodes
- population initialization in local and global search with candidate nodes
- define discretization rules and operators
- use expected diffusion value as a cost function

In these algorithms, the computational overhead is reduced by eliminating the Monte Carlo simulation and using expected diffusion value as cost function. Also, Meta-heuristic algorithms in IM problem using specific steps and by searching the problem space to find the optimal solution. These steps are:

1. Create an initial population
2. Update the initial population with a local search
3. Update local solutions with global search
4. Select best fitness solutions
5. Repeat steps 2 to 4 until reaching the stop condition
6. Update the final solution
7. Calculate the influence spread of the solution in step 6 with the diffusion models

A theoretical analysis of the IMP algorithms in social networks is shown in Table 5. Also, the explanation of each symbol of the time complexity column is shown in Table 6.

### 3.2 The meta-heuristic algorithms for the IMP in the social networks

In the paradigm of solving optimization problems by meta-heuristic algorithms, determine the solution space, and create an objective function is a necessity for each problem. In the following, evolutionary operations are used iteratively to find an optimal solution. In generally, in these algorithms by defining a cost function, the problem is modeled as an optimization problem, and methods like evolutionary optimization algorithms are applied to solve it. The use of meta-heuristic algorithms for the IMP was first presented by Jiang et al. by introducing both SA and SAEDV algorithms [50]. Then, the researchers have presented different algorithms based on the genetic algorithm to find influential nodes in the social networks [51, 60–63]. Also, other algorithms were presented for the IMP based on meta-heuristic algorithms using two approaches. These two approaches are: (1) the using of the influence spread estimation function as a cost function and (2) the using of the evolutionary rules in meta-heuristic algorithms. Accordingly, the first approach uses the Expected Diffusion Value (EDV) function or the Local Influence Estimator (LIE) function or Three Layer Comprehensive Influence Evaluation (TLCIE) or entropy of the worthiness function. Also, the second approach uses rules for discretization, local search, and global search. First, Gong et al. have used these two approaches to finding the influential nodes, and they have presented a meta-heuristic algorithm based on the particle swarm optimization algorithm called DPSO [52]. Given the acceptable and efficient results of the DPSO algorithm for the IMP, other meta-heuristic algorithms including the DDSE [53], DBA [54], ELDPPO [55], ELDPPO – BC [55], ELDPPO – CC [55], ADPSO [56], ADABC [57], GWIM [58] and DSFLA [59] algorithms from these two approaches to finding the influential nodes in social networks have used. In the following, these algorithms

**Table 5** The comparison the meta-heuristic algorithms for the IMP in social networks

Category	Algorithm name	Year	Diffusion model				Time complexity	Data set name	Data set scale			Comparable algorithm
			IC	WC	LT	SI			SIS	SIR	S	
Meta-heuristic algorithms	SA	2011	✓	✗	✗	✗	✗	Mobile, Epinions, Web, and Amazon	✗	✗	✓	NewGreedy, CGA, DD, Random, and SAEDV
	SAEDV	2011	✓	✗	✗	✗	✗	Mobile, Epinions, Web, and Amazon	✗	✗	✓	NewGreedy, CGA, DD, Random, SA, MSA, and SASH
	GNA	2015	✓	✗	✗	✗	✗	NetGR-QC and NetHEPT	✓	✓	✗	SA and NewGreedy
	DPSO	2016	✓	✓	✗	✗	✗	NetInfective, NetScience, NetGR-QC, and NetHEPT	✓	✓	✗	High Degree, PageRank, SAEDV, and CELF ++
	DDSE	2018	✓	✗	✗	✗	✗	NetScience, NetGR-QC, and NetHEPT	✓	✓	✗	CELF, SAEDV, PageRank, and High Degree
	DBA	2018	✓	✗	✗	✗	✗	HepTh, PGP SynRand, CondMat, Deezer, and Slashdot	✓	✓	✓	CELF, DFPSO, DDSE, and SSA
	ELDPSO	2019	✓	✗	✗	✗	✗	Karate, NetScience, Email, Blogs, CA-HepTh, and PGP	✓	✓	✗	General Greedy, CELF, SD, DD, High Degree, Betweenness centrality, and DPSO
	ELDPSO – BC		✓	✗	✗	✗	✗					
	ELDPSO-CC		✓	✗	✗	✗	✗					



Table 5 continued

Category	Algorithm name	Year	Diffusion model				Time complexity	Data set name	Data set scale			Comparable algorithm	
			IC	WC	LT	SI			SIS	SIR	S		M
	ADPSO	2019	✓	✗	✗	✗	✗	$O(k\bar{D}(nC + k\bar{D})g_{max} + (M + N))$	Email, Blogs, ca-HepTh, PGP, Astroph, and CondMat	✓	✓	✗	CELF, IMPC, DDSE, and SSA
	ADABC	2019	✓	✓	✗	✗	✗	$O(G_{max} * n * k^2 * P' * \bar{M})$	NetScience, Power, NetGR-QC, HEP-TH, and NewSites	✓	✗	✗	High Degree, Distance, DD, PageRank, CELF + , TLCIE, and DPSO
	GWIM	2020	✓	✗	✗	✗	✗	$O(N + g_{max}.n.V' + g_{max}.n.k.(\bar{D}).(\bar{D}^2))$	HAM, PGP, and AstroPh	✓	✓	✗	High Degree, K-shell, PageRank, Betweenness centrality, CELF + , DD, HC, DDSE, and SAEDV
	DSFLA	2020	✓	✗	✗	✗	✗	$O(\min k\bar{D} + (mn \log(mn) + mk\bar{D})g_{max})$	AstroPh, CondMat, Slashdot, Epinions, Eu-Email, and Stanford	✗	✓	✓	CELF, DFPO, DDSE, and SSA

**Table 6** The explanation of each symbol of the time complexity

Symbol	Explanation
$k$	The number of seed nodes
$R$	The number of Monte Carlo simulation
$N$	The number of nodes in the graph
$M$	The number of edges in the graph
$T$	The number of iterations
$R'$	The number of Monte Carlo simulation required by the StaticGreedy algorithm
$M'$	The average number of active edge in the snapshot obtained by sampling
$N'$	The number of nodes with LI value being 0 in the LIR algorithm
$pop$	The size of source node set in LGIM algorithm
$\overline{D}$	The average degree of the graph nodes
$T'$	The number of time steps during the label propagation
$k_{max}$	The maximum node degree
$n'$	The number of the candidate nodes
$m'$	The number of the candidate edges
$c$	the number of nodes in communities
$t_{\theta}$	Time complexity for LDAG construction for single node
$n_{\theta}$	Maximum possible nodes in a single LDAG
$m_{\theta}$	Maximum the possible edges in single LDAG
$P_{\theta}$	Maximum the possible path from a node
$O_v$	Max number of influence paths from a single node
$n_{vu}$	Max number of influence paths between any two nodes
$c'$	The number of parallelized process
$g_{max}$	The number of iterations in the meta-heuristic algorithms
$n$	The size of the population in the meta-heuristic algorithms
$m$	The number of the memplexes
$C$	The number of communities
$P'$	The size of elite node set
$\overline{M}$	The average of the sum of the number of three layers neighbors of the initial node set
$V'$	The number of $k$ nodes with the lower random value from population
$\overline{D}^2$	The second-order degree of nodes

are described. Figure 2, shows the timeline diagram of the meta-heuristic algorithms for the IMP in the social networks.

### 3.3 Simulated annealing (SA) algorithm

Simulated Annealing is an Intelligent Algorithm that simulates the process of metal annealing and optimizes the solutions of a number of NP-hard problems, e.g., the IMP in social networks [50]. The first meta-heuristic algorithm for the IMP is the SA algorithm by Jiang et al. based on the simulated annealing algorithm presented [50].

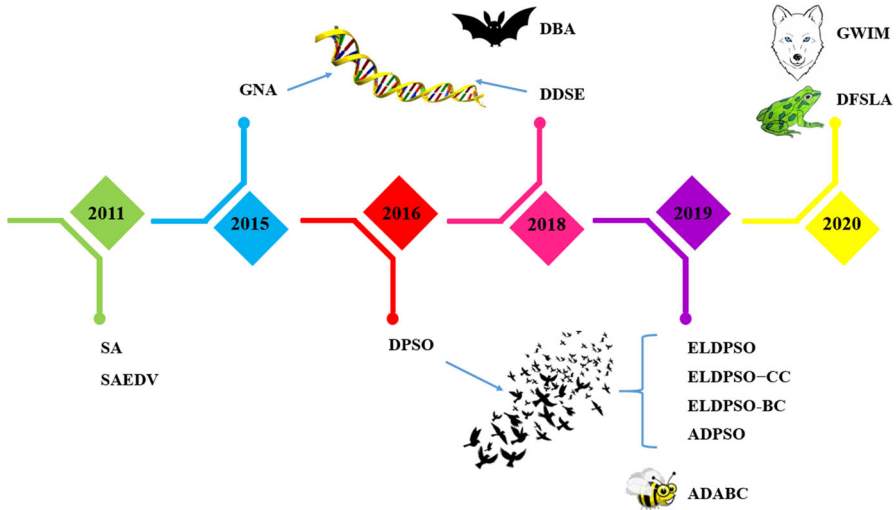


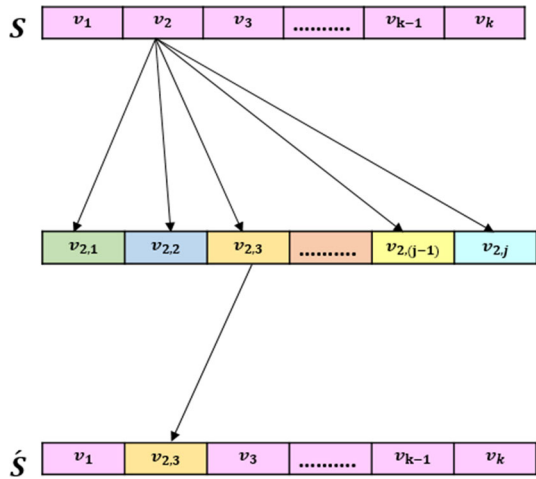
Fig. 2 The timeline of the meta-heuristic algorithms for the IMP in the social networks

This algorithm finds influential nodes with creating an initial solution and updating by neighbor solution and repeating this process. Since the neighbor solution is updating by the Monte Carlo simulation, the computational complexity of this method is high. This algorithm consists of five basic steps [50]: (1) creating an initial solution  $S$  and an initial temperature  $T = T_0$ , and the calculating of the cost function for an initial solution by  $\sigma(S)$  that  $S \subset V, |S| = k$ , (2) the searching of neighbor solutions for the current solution, the creating of a new solution  $S'$  (Fig. 3) and calculating of the cost function for current solution  $bss\sigma(S)$ , (3) If  $\Delta f < 0$ , the new solution is a better solution, so will replace the current solution  $i(\Delta f = \sigma(S) - \sigma(S'))$  and if  $\Delta f > 0$ , the new solution will replace the current solution with a possibility  $p_{i,j} = \exp(\frac{-\Delta f}{T})$ , (4) reducing of the temperature after a number of iterations for searching of the neighbor solutions ( $T = T - \Delta T$ ) and (5) stop of the algorithm when the temperature reaches the final temperature  $T_f$ .

The advantages of the SA algorithm are: (1) higher influence spread than DD, CGA, and NewGreedy algorithms, (2) lower running time than CGA and NewGreedy algorithms, (3) using the large-scale data sets, and (4) comparison with greedy and heuristic algorithms, but the iteration number of Monte Carlo simulation for computing of the influence spread is expensive and has led to the high running time in this algorithm.

Figure 3 depicts the local search steps around an S solution. In the SA method, the search is done to find the optimal solution in the problem space by creating an initial solution and updating the solution. Figure 3 shows the generation of an initial solution consisting of  $k$  nodes. In this solution, the second node is selected randomly. From the neighbors with a distance of one of these nodes, the third neighbor is randomly selected from the  $j$  neighbors and replaces the second node. Therefore, a local search is done around the S solution.

**Fig. 3** A simple example of creating the new neighbor solution in the SA algorithm



**3.4 SAEDV algorithm**

To improve the SA algorithm, SAEDV, and SASH presented. The expected diffusion value of the SAEDV algorithm as a cost function improves running time in the SA algorithm. The roulette wheel selection with high degree nodes of the SASH algorithm improves running time and influence spread in the SA algorithm. Jiang et al. have presented another meta-heuristic algorithm called SAEDV to improve the efficiency of the SA algorithm [50]. To compute  $\sigma(S)$  for a solution set  $S$ , the existing the IMP algorithms need  $R$  times simulations that This is computationally expensive and needs to run at least tens of thousands times to obtain an accurate estimation, which is very time-consuming. The SAEDV algorithm works just the same as the SA algorithm, except for computing of the cost function  $\sigma(S)$ . This algorithm uses the EDV function by Eq. (4) [50] to replace the diffusion simulations when computing  $\sigma(S)$  for a solution set  $S$  to cut down the computation cost and estimates the influence spread of a target set  $S$  in the SA algorithm, instead of using spreading simulations for  $R$  times.

$$EDV(S) = \sum_{i \in N_S^{(1)}/S} 1 - (1 - p)^{\tau(i)} \tag{4}$$

where,  $N_S^{(1)}$  represents the direct neighbors (one-hop) of, and  $i \in N_S^{(1)}/S$  represents that  $i$  belongs to  $N_S^{(1)}$  but not to  $S$ . Also,  $p$  is the value for activation probability, and  $\tau(i)$  denotes the number of links between node  $i$  and the initial node set  $S$ . The SAEDV algorithm is speedier than the SA algorithm but neglects some important real-world influence factors for real-world social network composed of countless nodes and relations between them that reduces the simulation accuracy of real information communication modes [57]. The advantages of this algorithm are the use of the EDV cost function and removal of Monte Carlo simulation, which leads to increased running time. Local search and creating many neighbor solution, low influence spread than

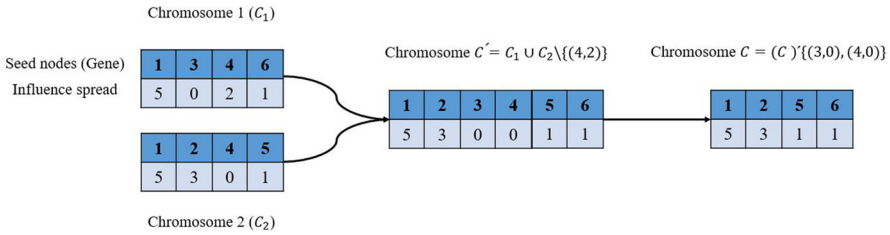


Fig. 4 A simple example for illustrating how crossover works in the GNA algorithm

the SA algorithm is one of the advantages of this algorithm. The disadvantages of the SA algorithm have been eliminated by improving local search in the SASH algorithm. One of the advantages of the SASH algorithm is the use of the roulette wheel selection to generate a local solution. But the number of iterations in the SASH algorithm is still the same as in the SAEDV algorithm. One of the disadvantages of this algorithm is the lack of use of special nodes, including nodes with a high degree of centrality to creating an initial solution.

### 3.5 GNA algorithm

In the GNA algorithm, the main goal is to combine the NewGreedy algorithm and the genetic algorithm. In the genetic algorithm, the influence of each node in the solutions is calculated based on the NewGreedy algorithm. High influence nodes are used to create a crossover solution. The use of this algorithm has led to a better local search in the genetic algorithm. Jiang et al. have presented the GNA algorithm based on the genetic meta-heuristic algorithm for the IMP [51]. This algorithm consists of five basic steps: (1) initialization, (2) update by the NewGreedy Algorithm, (3) selection step that selects the top chromosomes based on the influence of the chromosomes to participate in the crossover and uses the roulette wheel to selecting of the chromosomes. So the higher the influence of a chromosome, the higher the chance of being selected, (4) crossover step that removes useless node based on the influence spread of the seed nodes, and (5) mutation step which to find a better solution helps to the GNA. Figure 4 shows how crossover works in the GNA algorithm [51]. Figure 4 represents the generation steps of creating a crossover. Hence, two parents are initially selected and then the diffusion rate of each of the k nodes is calculated for the parents. After that, the nodes are sorted based on the diffusion rate and from 2k nodes, the number of k nodes with the highest diffusion rate is created as the crossover solution.

Disadvantages of this algorithm include not providing a solution for global search, increasing the number of iterations compared to the SA algorithm and not using conventional diffusion models.

### 3.6 DPSO algorithm

Gong et al. have presented the DPSO algorithm based on the particle swarm optimization algorithm to optimize the local influence criterion [52]. In the DPSO algorithm,

the DPSO algorithm uses the expected local influence estimation as a cost function. Also, discretization rules are used to update the created solution in this algorithm. In this algorithm, population initialization is based on high degree nodes. However, local search is useful for improve convergence. Moreover, they have introduced a degree based heuristic initialization strategy and a network-specific local search strategy to speed up the convergence. The EDV function only takes the neighbors of seed nodes into consideration, which cans only effectively tackle the situation with a small activation probability in the IC model. Therefore, the DPSO algorithm is designed a new function named Local Influence Estimator (LIE) to obtaining a suitable trade-off between running time and accuracy to compute the influence spread for set  $S$  by Eq. (5) [52].

$$LIE(S) = k + \left( 1 + \frac{1}{|N_S^{(1)}/S|} \sum_{u \in N_S^{(2)}/S} p_u^* d_u^* \right) \sum_{i \in N_S^{(1)}/S} \left( 1 - \prod_{(i,j) \in E, j \in S} (1 - p_{i,j}) \right) \quad (5)$$

where  $N_S^{(1)}$  represents the direct neighbors (one-hop) of set  $S$  and  $N_S^{(2)}$  represents the neighbors (two-hop) of set  $S$ . Also,  $p_u^*$  is the activation probability of node  $i$  under a diffusion model,  $d_u^*$  is the number of edges of a node  $u$  within  $N_S^{(1)}$  and  $N_S^{(2)}$ , which represents the number of activated probability for a node  $u$ , that generally  $d_u^* \leq p_u^*$ .

In the DPSO algorithm for the first time, the original form of the meta-heuristic algorithms that are used to solve continuous problems is modified. Accordingly, a number of specific rules have been used to discretization the solutions and to update them. To update the best solution in each iteration, each node is replaced with its neighbor. Figure 5 shows the discretization of the particles in the DPSO algorithm. Figure 5 indicates the local and global search steps in the PSO algorithm. This method includes three matrices with the dimensions  $n_{pop}$  row and  $k$  column. The matrices represent the initial population, velocity, and the best local solutions, respectively. Moreover, the vector of the best global solution represents the best solution up to the current iteration. This method uses discretization rules to update the initial population solutions. As seen in the figure,  $x_{ij}$  refers to the node in row  $i$  and column  $j$  in the initial population matrix. In this method, two steps are taken to update the matrix  $x$ , which are described below.

1. Initializing the row  $i$  and column  $j$  of the velocity matrix that is:

- A local intersection binary vector is initialized. If the node  $x_{ij}$  among the nodes in row  $i$  from the best local solution matrix the  $j$ th value of the local intersection vector is zero; otherwise, it is one.
- A global intersection binary vector is initialized. If the node  $x_{ij}$  among the nodes in the best global vector the  $j$ th value of the local intersection vector is zero; otherwise, it is one.
- The weighted sum of the local intersection, global, and velocity vectors in the  $i$ th row leads to the initialization of the velocity matrix.

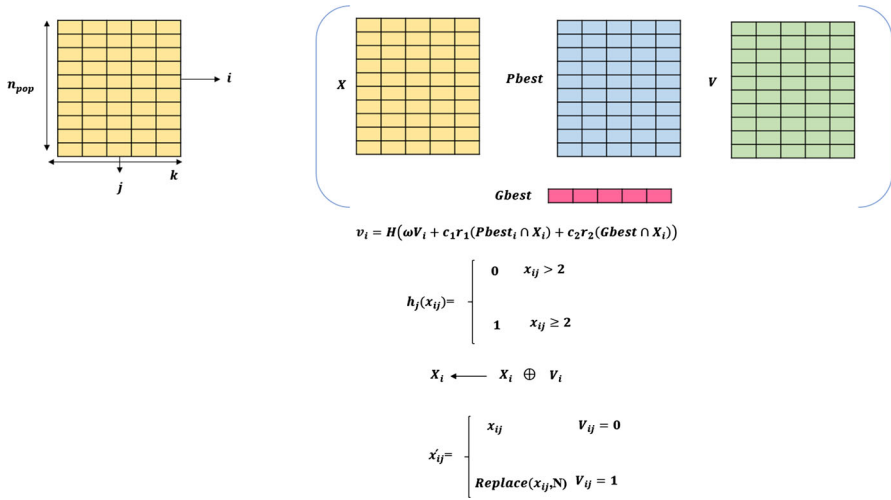


Fig. 5 Discretization of the particles in the DPSO algorithm

2. Initializing the initial population matrix

If  $v_{ij} < 2$ ,  $x_{ij}$  value is kept unchanged; otherwise,  $x_{ij}$  is randomly replaced by one of the graph nodes.

After updating the matrix  $x$ , the local population matrix, and global population vector update steps are performed. If row  $i$  of the matrix  $x$  has more fitness than the row  $i$  of the local population matrix, this row of the local population matrix is updated with the matrix  $x$ . Local search for each node in the best solution of the current iteration leads to the updating of the best global solution vector.

The advantages of this algorithm are the using of the discretization rules, the using of the LIE function as the cost function, and using two IC and WC models. The disadvantages of the DPSO algorithm are higher running time than the SAEDV algorithm, the lack of using the large-scale data set, and the lack of comparability with the new algorithms.

3.7 DDSE algorithm

Cui et al. have proposed the DDSE algorithm based on the memetic algorithm for the IMP [53]. In DDSE algorithm uses a degree descending search strategy to select and initialization Initial population, multination, and crossover populations. Creating an initial population with high-grade nodes in each iteration and updating the answer with a local search optimizes the selection of influential nodes. Also, using the expected diffusion value function leads to finding influential nodes in low running time. They have employed two techniques for improving efficiency [53]: (1) Degree-Descending Search strategy and (2) using of the objective function EDV. Firstly, they have proposed a degree-descending search strategy (DDS) that is capable of generating a node set whose influence spread is comparable to the High Degree algorithm. Then, based on DDS, they have developed the genetic algorithm that is capable of improving efficiency

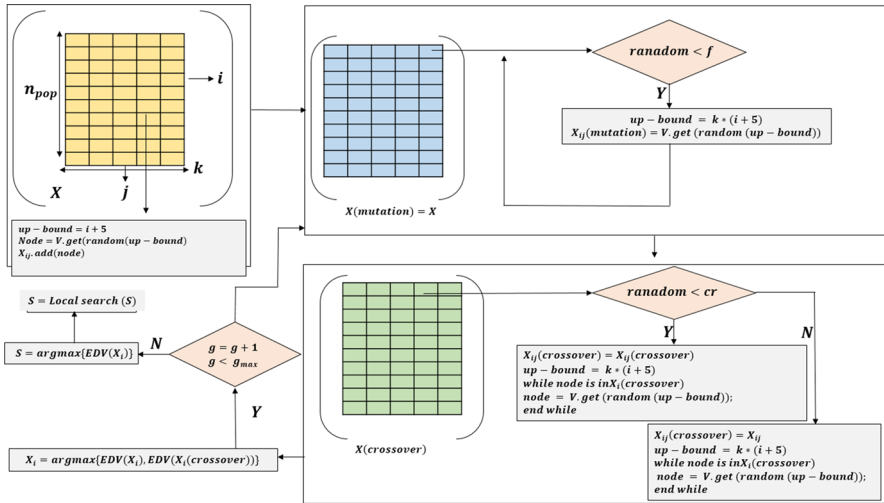


Fig. 6 General framework of the DDSE algorithm

by function EDV than greedy algorithms. In addition, a degree descending search strategy has been employed in initialization, mutation, and crossover operators[53]. Figure 6 shows the general framework of the DDSE algorithm. Figure 6 depicts the steps of the DDSE algorithm. In the DDSE algorithm, which is a modified version of the genetic algorithm, we have three matrices with the dimensions of  $n_{pop}$  row and  $k$  column. These three matrices represent the initial population, mutation population, and crossover population, respectively.

The following steps are taken to create the matrix  $x(mutation)$ :

First, row  $i$  and column  $j$  from  $x(mutation)$  are initialized with  $x_{ij}$ .

For each node of row  $i$  and column  $j$  of  $x(mutation)$ , a random number is generated. If  $random < f$ , this node is randomly replaced by one of the first nodes up to  $k \times (i + 5)$

Steps 1 and 2 are performed for each node of row  $i$  and column  $j$  of the matrix  $x(mutation)$

The following steps are performed to create the matrix  $x(crossover)$ :

1. For each node of row  $i$  and column  $j$  of  $x(crossover)$ , a random number is generated.
2. If  $random < cr$ ,  $x_{ij}(crossover)$  is initialized with  $x_{ij}(mutation)$ .
3. If  $random \geq cr$ ,  $x_{ij}(crossover)$  is initialized with  $x_{ij}$ .
4. Steps 1 and 2 are performed for each node of the row  $i$  and column  $j$  of the matrix  $x(mutation)$

If the fitness of the  $i$ th row of the crossover matrix is greater than the  $i$ th row of the matrix  $x$ , the value of the  $i$ th row of the matrix  $x$  is updated with the values of the row  $i$  of the crossover matrix.

The steps of creating matrices  $x(mutation)$  and  $x(crossover)$  as well as updating the matrix  $x$  are performed to reach the stop conditions with a certain number of



iterations. The row of the matrix  $x$  with the highest EDV is selected as the solution. For each node  $j$  in this solution, if replacing this node with any of the neighbors led to increase EDV, the node  $j$  is updated with its neighbor.

The main features of the DDSE algorithm are initialize of the solution and generating diversity randomly by high degree nodes, change the crossover, and mutation steps in the genetic algorithm. The advantages of the DDSE algorithm are: (1) using of the high degree nodes, (2) generating diversity randomly and simple, (3) lower running time than the SAEDV algorithm, and (4) using of the best solution updates after reaching stop condition, through local search. In the DPSO algorithm using discretization rules, the generated solutions are limited to the initial populations while in the DDSE algorithm, random change has helped to create diverse solution. In this algorithm, unlike the DPSO algorithm, local search is not performed in the iteration steps of the algorithm. Lack of comparison of the results of using the discretization rules and EDV cost function in comparison with DDSE algorithm is the most important drawback of this algorithm.

### 3.8 DBA algorithm

The presentation of the DPSO and the DDSE algorithms for the IMP led to the popularity and use of other meta-heuristic algorithms by researchers such as the bat algorithm. Tang et al. have presented the DBA algorithm based on the collective intelligence of the bat population for the IMP [54]. The DBA algorithm uses greedy local search and discretization rules based on network topology to create populations. Also, this algorithm uses random walk based on a combination of degree centrality and closeness centrality to update populations. In this algorithm, according to the evolutionary rules of the original bat algorithm, a probabilistic greedy-based local search strategy based on network topology has proposed. Then a CandidatesPool has generated according to the contribution of each node to the network topology to enhance the exploitation operation of DBA. A CandidatesPool maintaining potential influential nodes [54]. The main feature of this algorithm is to use discretization rules to generate the solution, local search, and random walk in the graph. Local search in the DBA algorithm compared with the DPSO algorithm consists of two major differences: (1) Local search has performed for all neighbors of the solution. In this search has used a probability 50% to replace of neighbors for every node, and (2) combination of degree and closeness centralities have used. As such, these two centralities are used for random walk in the graph and the selection of the candidate nodes. Figure 7 shows the general framework of the DBA algorithm. Figure 7 represents the steps of the BAT algorithm. Similar to the PSO method, this method uses discretization rules to initialize and update the initial velocity and population matrices. The two main differences in this method are how to update the velocity vector by considering the average of  $f_{max}$  and  $f_{min}$  solution instead of the constant value of 2 in the PSO algorithm and updating  $x_{ij}$  with the candidate pool nodes. These candidate pool nodes are selected by random walk in neighbors that have a higher degree of centrality and closeness than other nodes.

The DBA algorithm has three main advantages: (1) Comparison with recent algorithms such as DPSO and DDSE, (2) higher influence spread than DPSO and DDSE

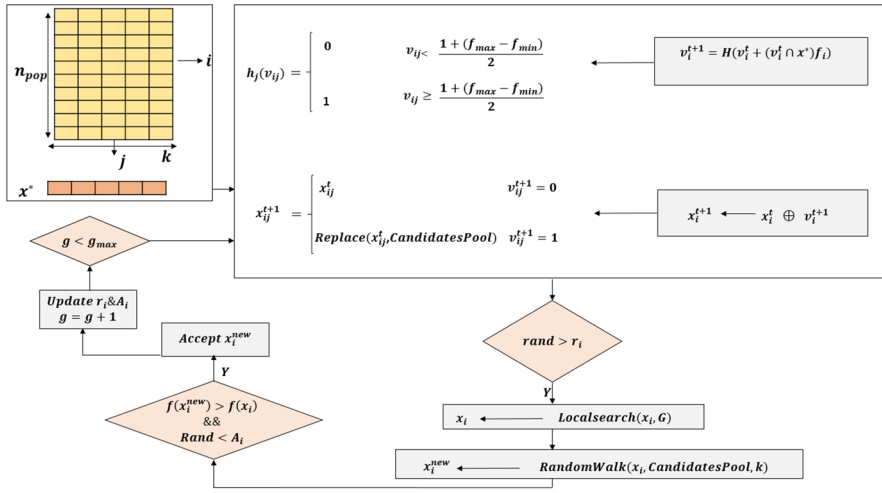


Fig. 7 The general framework of the DBA algorithm

algorithms, and (3) using the large-scale data set. Comparisons made in this algorithm can be concluded that using the LIE function compared to EDV leads to finding more influential nodes. The disadvantages of this approach are that it influence spread closer to DPSO in most datasets and longer running time than DPSO. In this algorithm, random step search and generation of candidate nodes for local search has led to an increase in running time. Therefore, using other algorithms such as reducing random walking space by removing nodes with low influence can help increase the speed of this algorithm.

### 3.9 ELDPSO, ELDPSO–BC, and ELDPSO–CC algorithms

Tang et al. have presented the ELDPSO algorithm by using a network topology-based local search strategy to enhance the local exploitation of the DPSO algorithm for the IMP [55]. In the ELPSOs method algorithm, the order of updating the nodes in the local search is based on the node degree centrality and closeness centrality measure. This algorithm improves DPSO results in terms of influence spread.

They have made the further exploration of the DPSO algorithm. They also have considered the effect of two other centralities including Betweenness and Closeness on the improvement of local search strategy while recombining the temporary seed nodes according to the two centralities separately and two other algorithms named ELDPSO–CC and ELDPSO–BC have presented also. As such, the using of these centralities for ordering nodes and updating the solution has led improve influence spread than the DPSO algorithm [55]. In fact, these algorithms work just the same as the DPSO algorithm, except for the local search these are somehow different. The main advantage of this algorithm is the improvement of results by focusing on local search. Updating to Local Search with best neighbor Instead of the first best neighbor, better results of LIE function than EDV are some of the advantages of this algorithm. Due

to the lack of comparison of the results of this algorithm with the DPSO algorithm, the effect of local search with high centrality nodes and updating with the best answer has not been determined. Lack of provide an algorithm to improve global search is another disadvantage of this algorithm.

### 3.10 ADPSO algorithm

Tang et al. have presented an adaptive discrete particle swarm optimization named ADPSO based on network topology by Label Propagation (LP) algorithm for the IMP [56]. In the ADPSO algorithm, the topology of graph nodes is used based on the feature of creating communities with the label propagation algorithm to the initial population. The purpose of this method is to initialize the population with nodes from different clusters. This algorithm improves DPSO results in terms of influence spread.

In the ADPSO algorithm, first, community structures are detected by the LP algorithm, then a dynamic encoding mechanism for particle individuals and discrete evolutionary rules for the swarm are conceived based on network community structure to identify the allocated number of influential nodes within different communities [56]. Also, to expand the seed nodes reasonably, a local influence preferential strategy presented to allocate the number of candidate nodes to each community according to its marginal gain [56]. Figure 8 shows the general framework of the ADPSO algorithm [56].

The main advantages of the ADPSO algorithm are: (1) higher influence spread than the DDSE and SSA algorithms, and (2) comparison with the recent algorithm. The disadvantages of the ADPSO algorithm are: (1) the lack of comparison with the DPSO algorithm, (2) higher running time than the IMPC algorithm, and (3) the lack of using the large-scale data set.

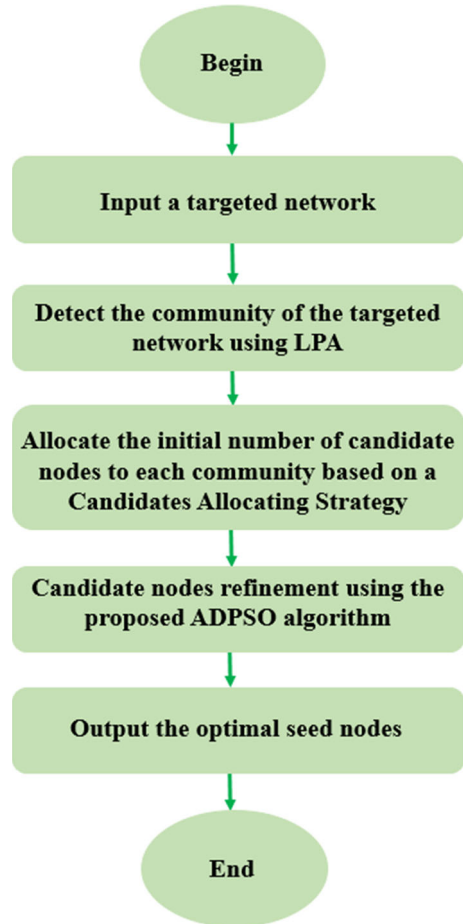
### 3.11 ADABC algorithm

Ma et al. have presented an adaptive discrete artificial bee colony algorithm (ADABC) to resolve the Three Layer Comprehensive Influence Evaluation (TLCIE) model for the IMP and create initialize population with high-degree nodes and the learning coefficient [57].

In this algorithm, the Comprehensive-Learning Guided (CLG) updating rules, the degree improvement initialization method, and the semi-abandonment scout bee strategy are incorporated to enhance the search ability [57]. Other meta-heuristic algorithms use the LIE function or the EDV function to estimate the influence spread. These two functions estimate the influence spread by one-hop and two-hop nodes of the seed set. For the first time in the ADABC algorithm, the estimation has computed using the neighbors of layer three of the initial nodes by [57]. Also, discretization rules such as DPSO and DBA algorithms have not used. Thus, in the local search and global search of the ADABC algorithm, each initial population member is updated by combining the best current solution with the solution from the nodes in the initial population.

Figure 9 shows the general framework of the ADABC algorithm [57]. The advantages of the ADABC algorithm are the higher influence spread and close running time

**Fig. 8** General framework of the ADPSO algorithm



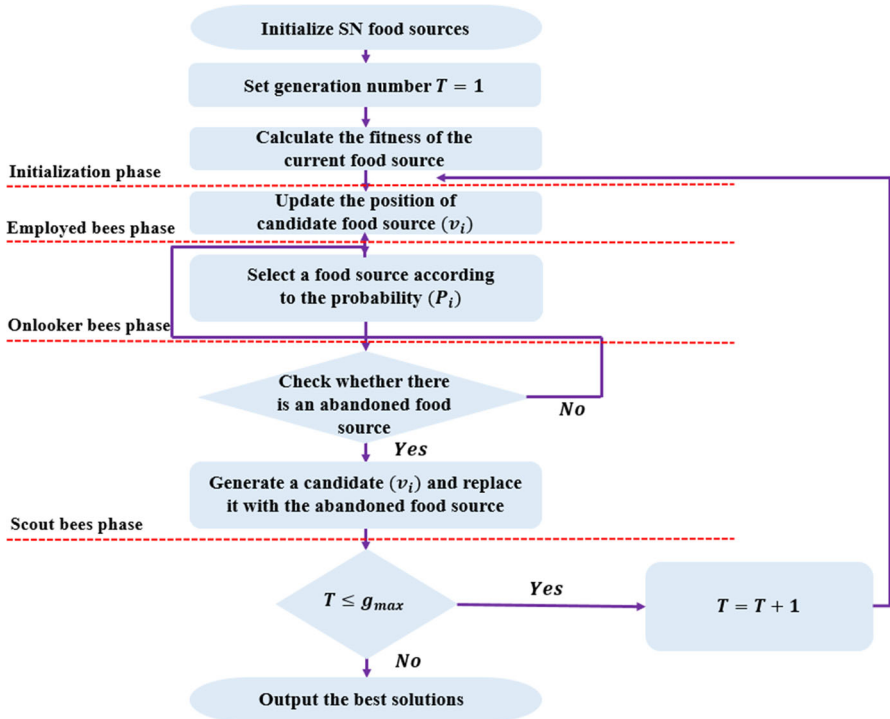


Fig. 9 General framework of the ADABC algorithm

than the DPSO algorithm. The disadvantages of this algorithm are the lack of using large-scale data set and the lack of comparison with other meta-heuristic algorithms that proposed recently.

### 3.12 GWIM algorithm

Zareie et al. have presented the GWIM algorithm based on the grey wolf algorithm for the IMP [58]. In the GWIM method, a cost function is introduced based on the distance between the nodes. This algorithm is continuously version in IM problem that improves results in term of influence spread and running time. In the GWIM algorithm, by using the concept of entropy a fitness function defined. Then, the IMP has solved using the grey wolf optimization algorithm. The main approaches in previous meta-heuristic algorithms are: local search to generate a neighbor solution, use of discretization rules, and use of the EDV or LIE as cost function, but in the GWIM meta-heuristic algorithm, these three approaches have changed completely. The main approaches in the GWIM algorithm are: using a specific cost function by Eq. (6) [58], lack of using local search, and using of the continuous variables instead of using graph nodes directly.

$$H(s) = - \sum_{v_j \in s} \frac{w(v_j)}{W(S)} * \ln \left( \frac{w(v_j)}{w(S)} \right) \quad (6)$$

where  $s$  shows the set of nodes located in the layer two and one neighbors of set  $S$  and  $W(S)$  is the sum of the worthiness of the nodes in  $s$  and is calculated by Eq. (7). In this Equation,  $w(v_j)$  is calculated by Eq. (8) [58].

$$W(S) = \sum_{v_j \in s} w(v_j) \quad (7)$$

$$w(v_j) = I(v_j) \cdot d_j \quad (8)$$

where  $d_j$  is the degree of node  $j$  and  $I(v_j)$  is calculated by Eq. (9) [58].

$$I(v_j) = \sum_{e_{ij} \in E \text{ and } v_i \in S} p_{ij} + \sum_{e_{ik} \in E \text{ and } e_{kj} \in E \text{ and } v_i \in S} p_{ik} \cdot p_{kj} \quad (9)$$

where  $p_{ij}$  is activation probability from  $v_i$  to  $v_j$ . The first section of Eq. (9) calculates the probability that the node receives the data of its neighbors, which are in set  $S$ ; it is considered 0 if the node has no neighbors in the set  $S$ . The second section of the Eq. (9), calculates the probability that the node receives the data from its layer two neighbors that are also in set  $S$ . The advantages of the GWIM algorithm are the higher influence spread and lower running time than the DDSE algorithm. Lack of compare the results of the proposed cost function in this algorithm with the LIE function, lack of use discretization rules, selection of  $k$  nodes from  $n$  nodes with sorting based on the random value of each node, are among the main disadvantages of this algorithm. Running time in the DDSE algorithm is more related to local search, while in this algorithm the cost function has more running time than the EDV function. Therefore, removing some nodes with low influence on the cost function can help reduce the running time of this algorithm. The selection of different  $p$  values for each data set is also one of the disadvantages of this algorithm and the proposed cost function.

### 3.13 DSFLA algorithm

Tang et al. have proposed an effective discrete shuffled frog-leaping algorithm (DSFLA) to solve the IMP [59]. The DSFLA method uses a combination of random and deterministic walking strategies in local search to create memes in created populations. This method has been used for better global search. Discretization rules based on graph topology and local search have led to better populations.

In this algorithm, a new encoding mechanism and discrete evolutionary rules considered based on network topology structure for virtual frog population. Also, to facilitate the global exploratory solution, a new local exploitation mechanism combining deterministic and random walk strategies detected to improve the sub optimal meme in the frog population. In generally, this algorithm used the memetic algorithm to generate a local search [59]. The main advantages of the DSFLA algorithm are: (1)

the using of the large-scale data set, (2) higher influence spread than DPSO, SSA, and DDSE algorithms, (3) comparison with the recent algorithm, and (4) lower running time than the DPSO algorithm. Three major disadvantages in the DSFLA algorithm are: (1) lower running time than the DDSE, SSA, and CELF algorithms, (2) lack of using discretization rules, and (3) lack of comparison with the DBA algorithm.

Table 7 shows the performance comparison of the meta-heuristic algorithms for the IMP. This table summarizes the advantages and disadvantages of each algorithm with their time complexity, diffusion model, and objective function. Also, Table 8 shows used data sets in each meta-heuristic algorithm for the IMP. In this table, column  $|V|$  refers to the number of the nodes in the graph, and column  $|E|$  refers to the number of the edges in the graph. The meta-heuristic algorithms for the IMP have compared with some baseline algorithms from each category. Table 9 shows compared algorithms with each meta-heuristic algorithm.

## 4 Discussion

In this section, we review and compare categorized algorithms. Two criteria, accuracy (influence spread) and efficiency (running time), are used to evaluate IM algorithms. Greedy algorithms guarantee the optimal approximation, but the optimal approximation guarantee depends on the number of repeats of the Monte Carlo simulation. That's why, in these algorithms, the process of selecting seed nodes takes several days and causes the inefficiency of these algorithms, even in small-scale datasets.

Heuristic algorithms have very good running time, but these algorithms do not guarantee optimal approximation. These algorithms examine only fixed features of the topology in any social network. But due to the dynamic structure of social networks, the topology feature may change from one network to another. That's why, this algorithm does not have an optimal influence spread on the selection of seed nodes.

Accuracy and efficiency in community-based algorithms depend on the quality of community detection [64, 65]. If the number of communities discovered is large, then the running time of the algorithm increases and the influence spread of seed node selection decreases. But these algorithms try to eliminate small and unsuitable communities, which has improved the running time of these algorithms. Also, these algorithms avoid the Rich Club phenomenon. That's why the selection of seed nodes from different communities. In the influence path-based algorithms, accuracy and efficiency depend on local criteria [65]. In these algorithms, optimal approximation is guaranteed according to local criteria.

Without knowing the features of the graph topology, meta-heuristic algorithms examine the IM problem as a search space. In these algorithms, without considering the features of the graph topology, the optimal approximation depends on the number of iterations, which increases the running time. If the features of nodes are examined in the social network topology, they converge with a low number of iterations and find the optimal solutions. Also, these algorithms use cost estimation functions for influence spread calculations, which eliminates the Monte Carlo simulation in the calculations, which makes this algorithm benefit from good efficiency. But these algorithms have running time problems in large-scale social networks.

**Table 7** Theoretical analysis of the meta-heuristic algorithms for the IMP in social networks

Algorithm name	Year	Diffusion model	Advantage	Disadvantage
SA	2011	IC	Using large-scale data sets, comparison with some of greedy and heuristic algorithms, higher influence spread than the DD, CGA, and NewGreedy algorithms, lower running time than CGA and NewGreedy algorithms	The high number of iterations
SAEDV	2011	IC	Using the EDV function for the first time and improve the running time of the SA algorithm	Lower influence spread than the SA algorithm
GNA	2015	IC	Higher influence spread and lower running time than the SA algorithm	The use of a higher number of iterations than the SA algorithm and the lack of using large-scale data set
DPSO	2016	IC, WC	The use of discretization rules and the use of the LIE function for the first time	The higher running time than the SAEDV algorithm, the lack of using the large-scale data set and the lack of comparability with the new meta-heuristic algorithms
DDSE	2018	IC	Lower running time than the SAEDV algorithm, changes to the original version of the Genetic algorithm and algorithm simplicity	The lack of comparison with the new algorithms and the lack of using large-scale data set

The best influence spread is for greedy algorithms (in the high iteration of Monte Carlo simulations) but these algorithms are inefficient for selecting seed nodes. It is the fastest running time for heuristic algorithms, but in general the influence path-based and meta-heuristic algorithms are a trade-off between influence spread and efficiency.

we use 2 networks (PGP dataset and Gr-QC dataset) to evaluate the influence spread and running time of the DegreeDiscount and LIR algorithms of the heuristic algorithms, the TI-SC and PHG algorithms of the community-based algorithms, and



Table 7 continued

Algorithm name	Year	Diffusion model	Advantage	Disadvantage
DBA	2018	IC	The using the large-scale data set, the using combination of degree and closeness centralities and higher influence spread than the DPSO and DDSE algorithms	The higher running time than the DPSO and DDSE algorithms
ELDPSO ELDPSO – BC ELDPSO-CC	2019	IC	The higher influence spread than the DPSO algorithm, using three centralities: degree, closeness, and betweenness in local search	The lack of comparison with other meta-heuristic algorithms, the higher running time than the DSPO algorithm and the lack of using large-scale data set
ADPSO	2019	IC	The using clustering of nodes, the using of the large-scale data set, and the higher influence spread than the DDSE algorithm	The lack of comparison with the DPSO algorithm and higher running time than the IMPC algorithm
ADABC	2019	IC, WC	The higher influence spread and close running time than the DPSO algorithm, use the new influence estimation function	The lack of comparison with the large-scale data set, the lack of comparison with other meta-heuristic algorithms
GWIM	2020	IC	Algorithm simplicity, the higher influence spread and lower running time than the DDSE algorithm, using a specific cost function,	The lack of comparison with the large-scale data set, the lack of comparison with other meta-heuristic algorithms, and the lack of using discretization rules
DSFLA	2020	IC	The algorithm simplicity, the higher influence spread than DPSO and DDSE algorithms, and using large-scale data set	Lack of comparison with the DBA algorithm and lack of using discretization rules

**Table 8** Statistical properties used data sets for the meta-heuristic algorithms in the IMP

Data set name	Meta heuristic algorithms for the IMP												
	V	E	SA	SAEDV	GNA	DPSO	DDSE	DBA	ELDPSO ELDPSO-CC ELDPSO-BC	ADPSO	ADABC	GWIM	DSFLA
Karate	34	78	-	-	-	-	-	-	✓	-	-	-	-
NetScience1	379	914	-	-	-	-	-	-	✓	-	-	-	-
NetInfective	410	17,298	-	-	-	✓	-	-	-	-	-	-	-
Email	1133	5451	-	-	-	-	-	-	✓	✓	-	-	-
NetScience2	1461	2742	-	-	-	-	✓	-	-	-	-	-	-
NetScience3	1589	2742	-	-	-	✓	-	-	-	✓	-	-	-
HAM	2426	16,631	-	-	-	-	-	-	-	-	✓	-	-
Blogs	3982	6803	-	-	-	-	-	-	✓	✓	-	-	-
Power	4941	6594	-	-	-	-	-	-	-	✓	-	-	-
NetGR-QC1	5242	14,496	-	-	✓	-	-	-	-	✓	-	-	-
NetGR-QC2	5242	28,980	-	-	-	-	✓	-	-	-	-	-	-
HEP-TH	8361	15,751	-	-	-	-	-	-	-	-	✓	-	-
CA-HepTh	9877	25,998	-	-	-	-	-	✓	✓	✓	-	-	-
PGP	10,680	24,316	-	-	-	-	✓	✓	✓	-	✓	-	-
SynRand	15,000	56,152	-	-	-	-	✓	✓	-	-	-	-	-
NetHEPT	15,233	58,891	-	-	✓	✓	✓	-	-	-	-	-	-
AST	18,771	198,050	-	-	-	-	-	-	-	-	✓	-	-
AstroPh	18,772	198,110	-	-	-	-	-	-	-	✓	-	-	✓
CondMat	23,133	186,936	-	-	-	-	-	✓	-	✓	-	-	✓
NewSites	27,917	206,259	-	-	-	-	-	-	-	✓	-	-	-

**Table 8** continued

Data set name	Statistic characteristics		Meta heuristic algorithms for the IMP												
	V	E	SA	SAEDV	GNA	DPSO	DDSE	DBA	ELDPSO ELDPSO-CC ELDPSO-BC	ADPSO	ADABC	GWIM	DSFLA		
Deezer	54,573	846,915	-	-	-	-	-	✓	-	-	-	-	-		
Slashdot	77,360	905,468	-	-	-	-	✓	-	-	-	-	-	✓		
Epinions	75,879	508,837	✓	✓	-	-	-	-	-	-	-	-	✓		
Mobile	95.7 k	1.33 M	✓	✓	-	-	-	-	-	-	-	-	-		
Amazon	262 K	1.23 M	✓	✓	-	-	-	-	-	-	-	-	-		
Eu-Email	265,214	420,045	-	-	-	-	-	-	-	-	-	-	✓		
Stanford	281,903	2,312,497	-	-	-	-	-	-	-	-	-	-	✓		
Web	345 K	421.6 K	✓	✓	-	-	-	-	-	-	-	-	-		

**Table 9** Compared algorithms with the meta-heuristic algorithms for the IMP

Baseline algorithms	Meta heuristic algorithms for the IMP										
	SA	SAEDV	GNA	DPSO	DDSE	DBA	ELDPSO ELDPSO – CC ELDPSO–BC	ADPSO	ADABC	GWIM	DSFLA
CELF	-	-	-	-	✓	✓	✓	✓	-	-	✓
DPSO	-	-	-	✓	-	✓	✓	-	✓	-	✓
DDSE	-	-	-	-	✓	✓	-	✓	-	✓	✓
SSA	-	-	-	-	-	✓	-	✓	-	-	✓
High Degree	-	-	-	✓	✓	-	✓	-	✓	✓	-
k-shell	-	-	-	-	-	-	-	-	-	✓	-
PageRank	-	-	-	✓	✓	-	-	-	✓	✓	-
BC	-	-	-	-	-	-	✓	-	-	✓	-
CELF++	-	-	-	✓	-	-	-	-	✓	✓	-
DD	✓	✓	-	-	-	-	-	-	✓	✓	-
HC	-	-	-	-	-	-	-	-	-	✓	-
SAEDV	✓	✓	-	✓	✓	-	-	-	-	✓	-
Distance	-	-	-	-	-	-	-	-	✓	-	-
IMPC	-	-	-	-	-	-	-	✓	-	-	-
General Greedy	-	-	-	-	-	-	✓	-	-	-	-
SD	-	-	-	-	-	-	✓	-	-	-	-
Diffusion Degree	-	-	-	-	-	-	✓	-	-	-	-
NewGreedy	✓	✓	✓	-	-	-	-	-	-	-	-
SA	✓	✓	✓	-	-	-	-	-	-	-	-
CGA	✓	✓	✓	-	-	-	-	-	-	-	-
Random	✓	✓	✓	-	-	-	-	-	-	-	-

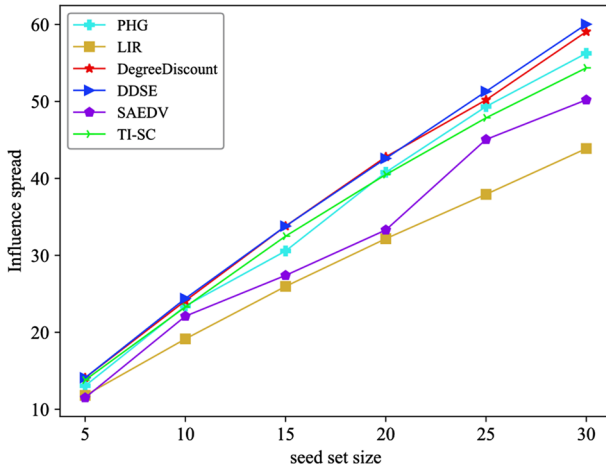


Fig. 10 Influence spread comparison in PGP dataset under independent cascade model

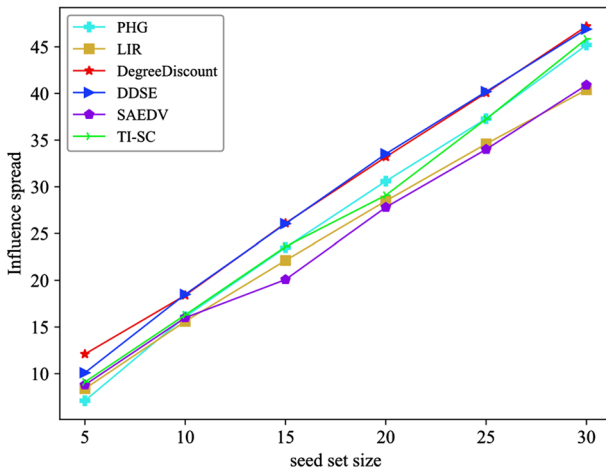


Fig. 11 Influence spread comparison in Gr-QC dataset under independent cascade model

the DDSE and SAEDV algorithms of the meta-heuristic algorithms in Figs. 10, 11, 12, 13. Due to Greedy and the influence path-based algorithms lack of efficiency, its results are not reported for PGP and Gr-QC. Figures 10 and 11 shows the amount of the influence spread of algorithms under independent cascade models where the x-axis indicates the number of seed nodes and the y-axis indicates the overall influence spread. In Fig. 10, DDSE is better than DegreeDiscount, PHG, TI-SC, SAEDV, and LIR on the PGP network. In Fig. 10, DegreeDiscount and DDSE algorithms have the same influence spread value. In Figs. 12 and 13, it is obvious that the running time of SAEDV and is lower other algorithms. However, it is obvious that SAEDV has the lowest quality in terms of influence spread.

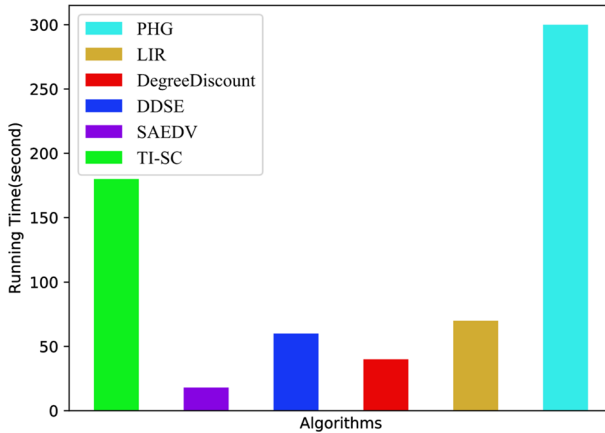


Fig. 12 Running times on PGP dataset (All times are in seconds)

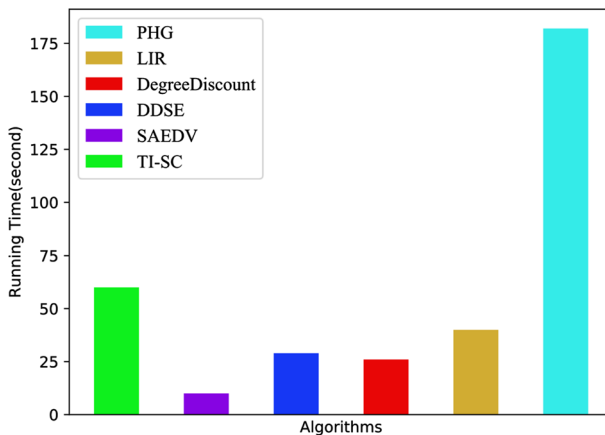


Fig. 13 Running times on Gr-QC dataset (All times are in seconds)

## 5 Conclusion and future work

A certain number of social network users to make the most effect on other users in the shortest possible time are selected. Due to the challenges of this problem, different algorithms by the researchers to find influential nodes have been presented. This paper first defines the IMP. Then, introduces diffusion models and describes the models that are most commonly used in IMP. In the following, a comprehensive categorize based on the methods of the presented algorithms for solving the IMP is proposed. Then we review, compare, and express the advantages and disadvantages of a category of these algorithms called meta-heuristic algorithms. In addition, the main feature of these algorithms is the use of diffusion estimation functions as cost function. Also, their main focus is on the initial population, the use of local and global searches, and the use of operators for discretization rule.

Research about a meta-heuristic algorithm may be carried out in the future, in the following directions:

Time and budget are two important criteria in maximizing penetration, so we will use these two criteria as multi-objective and constraint meta-heuristics algorithm in the cost function. In the real world, social networks are dynamic and the structure of the network changes over time, so the study of meta-heuristic methods in dynamic and temporary graphs is suggested. Meta-heuristic algorithms do not avoid the Rich Club phenomenon, so in selecting seed nodes, approaches should be used that these algorithms avoid the Rich Club phenomenon.

## References

1. Kempe D, Kleinberg J, Tardos É (2003) Maximizing the spread of influence through a social network. In: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining
2. Aghaee Z, Kianian S (2020) Efficient influence spread estimation for influence maximization. *Soc Netw Anal Min* 10(1):1–21
3. Beni HA et al (2020) IMT: selection of top-k nodes based on the topology structure in social networks. In: 2020 6th international conference on web research (ICWR). IEEE
4. Aghaee Z et al (2020) A heuristic algorithm focusing on the rich-club phenomenon for the influence maximization problem in social networks. In: 2020 6th International Conference on Web Research (ICWR). IEEE
5. Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining
6. Li W et al (2020) Three-hop velocity attenuation propagation model for influence maximization in social networks. *World Wide Web* 23(2):1261–1273
7. Sumith N, Annappa B, Bhattacharya S (2018) Influence maximization in large social networks: Heuristics, models and parameters. *Futur Gener Comput Syst* 89:777–790
8. Sanatkar MR (2020) The dynamics of polarized beliefs in networks governed by viral diffusion and media influence. *Soc Netw Anal Min* 10(1):1–21
9. Saxena B, Kumar P (2019) A node activity and connectivity-based model for influence maximization in social networks. *Soc Netw Anal Min* 9(1):40
10. Peng S et al (2018) Influence analysis in social networks: A survey. *J Netw Comput Appl* 106:17–32
11. Tang J, Tang X, Yuan J (2018) An efficient and effective hop-based approach for influence maximization in social networks. *Soc Netw Anal Min* 8(1):10
12. Chang B et al (2018) Study on information diffusion analysis in social networks and its applications. *Int J Autom Comput* 15(4):377–401
13. Banerjee S, Jenamani M, Pratihari DK (2018) A survey on influence maximization in a social network. arXiv preprint [arXiv:1808.05502](https://arxiv.org/abs/1808.05502)
14. Sun J, Tang J (2011) A survey of models and algorithms for social influence analysis. *Social network data analytics*. Springer, pp 177–214
15. Li M et al (2017) A survey on information diffusion in online social networks: Models and methods. *Information* 8(4):118
16. Pei S, Makse HA (2013) Spreading dynamics in complex networks. *J Stat Mech: Theory Exp* 2013(12):P12002
17. Pastor-Satorras R et al (2015) Epidemic processes in complex networks. *Rev Mod Phys* 87(3):925
18. Samadi N, Bouyer A (2019) Identifying influential spreaders based on edge ratio and neighborhood diversity measures in complex networks. *Computing* 101(8):1147–1175
19. Zafarani R et al (2014) Information diffusion in social media. *Social Media mining: an introduction*. NP, Cambridge UP
20. Rui X et al (2019) A reversed node ranking approach for influence maximization in social networks. *Appl Intell* 49(7):2684–2698

21. Singh SS et al (2020) IM-SSO: Maximizing influence in social networks using social spider optimization. *Concurr Comput Pract Exp* 32(2):e5421
22. Shang J et al (2017) CoFIM: a community-based framework for influence maximization on large-scale networks. *Knowl-Based Syst* 117:88–100
23. Leskovec J et al (2007) Cost-effective outbreak detection in networks. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*
24. Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*
25. Goyal A, Lu W, Lakshmanan LV (2011) Celf++ optimizing the greedy algorithm for influence maximization in social networks. In: *Proceedings of the 20th international conference companion on World wide web*
26. Cheng S et al (2013) Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In: *Proceedings of the 22nd ACM international conference on Information and knowledge management*
27. Heidari M, Asadpour M, Faili H (2015) SMG: Fast scalable greedy algorithm for influence maximization in social networks. *Phys A* 420:124–133
28. Zhang J-X et al (2016) Identifying a set of influential spreaders in complex networks. *Sci Rep* 6:27823
29. Berahmand K, Bouyer A, Samadi N (2019) A new local and multidimensional ranking measure to detect spreaders in social networks. *Computing* 101(11):1711–1733
30. Liu D et al (2017) A fast and efficient algorithm for mining top-k nodes in complex networks. *Sci Rep* 7:43330
31. Luo Z-L et al (2012) A pagerank-based heuristic algorithm for influence maximization in the social network. *Recent progress in data engineering and internet technology*. Springer, pp 485–490
32. Ahajjam S, Badir H (2018) Identification of influential spreaders in complex networks using HybridRank algorithm. *Sci Rep* 8(1):1–10
33. Qiu L et al (2019) LGIM: A global selection algorithm based on local influence for influence maximization in social networks. *IEEE Access*
34. Tang Y, Xiao X, Shi Y (2014) Influence maximization: Near-optimal time complexity meets practical efficiency. In: *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*
35. Tang Y, Shi Y, Xiao X (2015) Influence maximization in near-linear time: A martingale approach. In: *Proceedings of the 2015 ACM SIGMOD international conference on management of data*
36. Aghaee Z, Kianian S (2020) Influence maximization algorithm based on reducing search space in the social networks. *SN Appl Sci* 2(12):1–14
37. Wang Y et al (2010) Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*
38. Zhao Y, Li S, Jin F (2016) Identification of influential nodes in social networks with community structure based on label propagation. *Neurocomputing* 210:34–44
39. Qiu L et al (2019) PHG: a three-phase algorithm for influence maximization based on community structure. *IEEE Access* 7:62511–62522
40. Singh SS et al (2019) C2IM: community based context-aware influence maximization in social networks. *Phys A* 514:796–818
41. Bouyer A, Ahmadi H (2018) A new greedy method based on cascade model for the influence maximization problem in social networks. *J Inf Commun Technol* 10(37):85–100
42. Banerjee S, Jenamani M, Pratihari DK (2019) CombBIM: a community-based solution approach for the Budgeted Influence Maximization Problem. *Expert Syst Appl* 125:1–13
43. Beni HA, Bouyer A (2020) TI-SC: top-k influential nodes selection based on community detection and scoring criteria in social networks
44. Kimura M, Saito K (2006) Tractable models for information diffusion in social networks. In: *European conference on principles of data mining and knowledge discovery*. Springer, Berlin
45. Chen W, Yuan Y, Zhang L (2010) Scalable influence maximization in social networks under the linear threshold model. In: *2010 IEEE international conference on data mining*. IEEE
46. Chen W, Wang C, Wang Y (2010) 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*
47. Goyal A, Lu W, Lakshmanan LV (2011) Simpath: An efficient algorithm for influence maximization under the linear threshold model. In: *2011 IEEE 11th international conference on data mining*. IEEE



48. Wang C, Chen W, Wang Y (2012) Scalable influence maximization for independent cascade model in large-scale social networks. *Data Min Knowl Disc* 25(3):545–576
49. Kim J, Kim S-K, Yu H (2013) Scalable and parallelizable processing of influence maximization for large-scale social networks? In: 2013 IEEE 29th international conference on data engineering (ICDE). IEEE
50. Jiang Q et al (2011) Simulated annealing based influence maximization in social networks. In: Twenty-fifth AAAI conference on artificial intelligence
51. Tsai C-W, Yang Y-C, Chiang M-C (2015) A genetic newgreedy algorithm for influence maximization in social network. In: 2015 IEEE international conference on systems, man, and cybernetics. IEEE
52. Gong M et al (2016) Influence maximization in social networks based on discrete particle swarm optimization. *Inf Sci* 367:600–614
53. Cui L et al (2018) DDSE: a novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks. *J Netw Comput Appl* 103:119–130
54. Tang J et al (2018) Maximizing the spread of influence via the collective intelligence of discrete bat algorithm. *Knowl-Based Syst* 160:88–103
55. Tang J et al (2019) Identification of top-k influential nodes based on enhanced discrete particle swarm optimization for influence maximization. *Phys A* 513:477–496
56. Tang J et al (2019) An adaptive discrete particle swarm optimization for influence maximization based on network community structure. *Int J Modern Phys C (IJMPC)* 30(06):1–21
57. Ma L, Liu Y (2019) Maximizing three-hop influence spread in social networks using discrete comprehensive learning artificial bee colony optimizer. *Appl Soft Comput* 83:105606
58. Zareie A, Sheikahmadi A, Jalili M (2020) Identification of influential users in social network using gray wolf optimization algorithm. *Expert Syst Appl* 142:112971
59. Tang J et al (2020) A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks. *Knowl-Based Syst* 187:104833
60. Krömer P, Nowaková J (2017) Guided genetic algorithm for the influence maximization problem. In: international computing and combinatorics conference. Springer, Berlin
61. Weskida M, Michalski R (2016) Evolutionary algorithm for seed selection in social influence process. In: 2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). IEEE
62. da Silva AR et al (2018) Influence maximization in network by genetic algorithm on linear threshold model. In: international conference on computational science and its applications. Springer, Berlin
63. Bucur D, Iacca G (2016) Influence maximization in social networks with genetic algorithms. In: European conference on the applications of evolutionary computation. Springer, Berlin
64. Bouyer A, Roghani H (2020) LSMD: a fast and robust local community detection starting from low degree nodes in social networks. *Futur Gener Comput Syst* 113:41–57
65. Taheri S, Bouyer A (2020) Community detection in social networks using affinity propagation with adaptive similarity matrix. *Big Data* 8(3):189–202

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.