Journal of Network and Innovative Computing
ISSN 2160-2174 Volume 1 (2013) pp. 163 - 172

Evolutionary Approach for k-Max Influence Problem

Hema Banati[1] and Monika Bajaj[2]

[1] Dyal Singh College, University of Delhi,
New Delhi, India
banatihema@hotmail.com

[2] Department of Computer Science, University of Delhi,
New Delhi, India
mbajaj48@gmail.com

Abstract: **In the cut throat competitive environment of e-markets, research is currently being directed towards developing marketing strategies to promote products in minimum cost. Companies aim for maximum influence of their promotional activities at minimalistic cost. The problem of Maximizing influence spread with the limited seeding budget (*k*) in large network is denoted as *k-Max-Influence* problem and proven to be NP-Hard. Various methods have been proposed to tackle this problem. Although these methods are able to find the best seeds but suffer from high computation cost on estimating the influence function or require global knowledge of the network. This paper explores the viability of evolutionary algorithms for this problem vis-à-vis the contemporary greedy approach. It compares two prominent evolutionary algorithms i.e. Differential Evolution (DE) and Firefly (FA) for their suitability to *k-Max-Influence* problem. Experimental study was conducted on Epinions, Wiki-Vote, Slashdot, NetHEPT and NetPHY datasets. The results revealed that both evolutionary approaches DE and FA perform better as compared to Greedy approach with respect to maximum influence incurred as well as gain achieved by increasing the value of *k*. Amongst the evolutionary approaches FA outperform DE in all cases. The results show that FA maintains the consistency in its results and has higher probability to score over DE and Greedy.**

Keywords: Evolutionary Algorithm, E-Marketing, k-Max Influence Problem

I. Introduction

E-commerce has witnessed a significant growth in recent years. Industry has identified and acknowledged the potential of technology for online business. However, the actual business is usually preceded by active e-marketing strategies which involve promoting new products to prospective customers by ways of advertisements or distribution of "complimentary" / discounted products or services. The magnitude of online world makes such promotions both economically and practically non feasible. Research is therefore currently directed towards identifying techniques with a wider influence at low cost [1]-[4]. One of them is the use of social networking sites for product promotion. The past few years has seen a boom in social networking sites (SNS). SNS allow users to share their personal opinions and suggestions with their electronic peers. Amalgamation of millions of online users cutting across continents makes these SNS a powerful marketing platform to capture the attention of a wide range of audience on a small fraction of marketing budget. The sharing of experiences and opinions about a product among electronic peers facilitates the dissemination of both positive and negative electronic word of mouth (e-WOM) in competitive environment of e-commerce [5]. e-WOM has a great influence on the purchasing decision of consumer and forms a rich source of information for organizations to strategize their e-marketing plans.

A distinct e-marketing strategy that utilizes the connectivity of users on social networking sites to promote the products was developed [6]. It consisted of three phases. The strategy extracts the relevant product features and user interest towards those features to cluster users in accordance to their preferences. The key concept of this strategy is to promote products to initial users called *seeds* that can maximize the overall influence over a selected segment. These users then share their experiences with (some of) their friends who then pass on the word to their circle. Thus the message takes on the form of "virus" that spread through contact with others and over a period of time it covers portions of the social network far beyond the friend network of initial seeds. This network is tapped in third phase i.e. product promotion. The decision regarding the number of seeds that should be employed for marketing depends upon the marketing budget. Maximizing influence spread with the limited seeding budget i.e. *k* in large network is denoted as *k-Max-Influence* problem and proven to be NP-Hard [3]. Various efforts [1],[7]-[9] have been put in this direction. These methods succeed in finding the best seeds but suffer from high computation cost on estimating the influence function or require global knowledge of the network.

The paper studies the applicability of evolutionary approach for the same. Evolutionary Algorithms have achieved reasonable success at providing good feasible solutions to complex optimization problems [10],[11]. These algorithms have ability to cope with local optima by maintaining, recombining and comparing several candidate solutions simultaneously. This paper explores the viability of evolutionary algorithms for this problem vis-à-vis the contemporary greedy approach. It compares two prominent evolutionary algorithms i.e. Differential Evolution (DE) and

Firefly (FA) for their suitability to *k-Max-Influence* problem. The performance of each algorithm is compared with Greedy approach and evaluated with respect to maximum number of nodes influenced with *k* seed size and the marginal gain incurred by increasing the value of *k*.

The organization of the paper is as follows. Section 2 explains the E-marketing model. Section 3 gives an overview of *k-Max-Influence* problem followed by existing work carried out in this direction and models used for information diffusion. The following sections describes the Greedy algorithm, DE and FA and for *k-Max-Influence* problem. Section 7 and 8 outline the experimental set-up and analysis of results followed by conclusion in section 9.

II. E-Marketing Model

Web users often share their experiences and opinions on various products through online social media. This sharing of experiences works as electronic word of mouth (e-WOM) publicity for a product in the Internet world and plays an important role in decision making criterion for prospective customers. The e-marketing model shown in fig 1 takes advantage of quick spread behavior of e-WOM along with attraction mechanism of firefly algorithm for market campaigning. The whole model is divided into three phases Market analysis, Market Segmentation and Product Promotion. The functionality of each phase is depicted in fig 2.

A. *Phase I: Market Analysis*

The first phase analyzes the market trend in terms of most relevant product features. The market trend with respect to user needs changes due to technological advancements with due course of time. For example a decade ago users generally considered the size of mobile while purchasing it but now their taste has been shifted towards its camera pixels or audio quality. The changing trends with respect to relevant product features that are of interest to prospective customers is captured by mining the opinions/reviews given by the users. Product features are usually referred as nouns or noun phrases in review sentences. Each review is parsed through POS tagger and specific linguistic rules mentioned in [12] are then applied to extract the relevant product features. The method generates various noun terms and considering all these terms for further processing is cumbersome. Thus this dimensionality is reduced by applying a feature selection approach using firefly algorithm FA_RSAR proposed in [14]. Subsequently user profile based on the likes and dislikes of the user towards a product feature is generated. Users are classified into two categories i.e. active users and passive users. Active users continuously share their experiences and opinions for a product. The methodology proposed in [12] is applied to generate profile for active users. Passive users only read and consider reviews while purchasing a product. Their interest is therefore implicitly captured by analyzing the time spent on each product review [14].

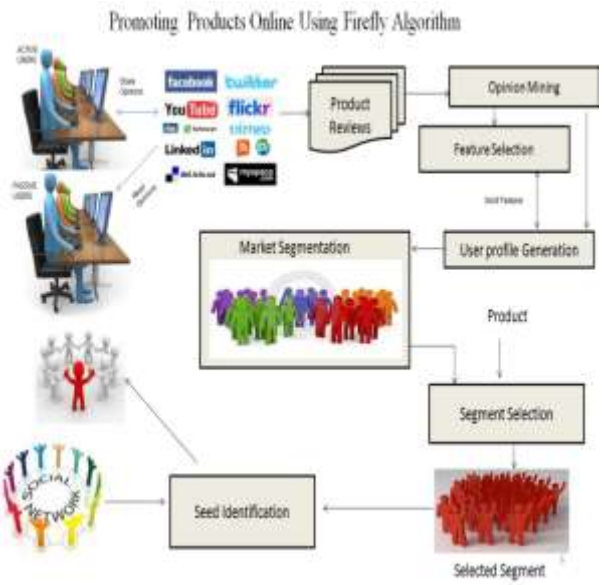


Figure1 E-Marketing Model

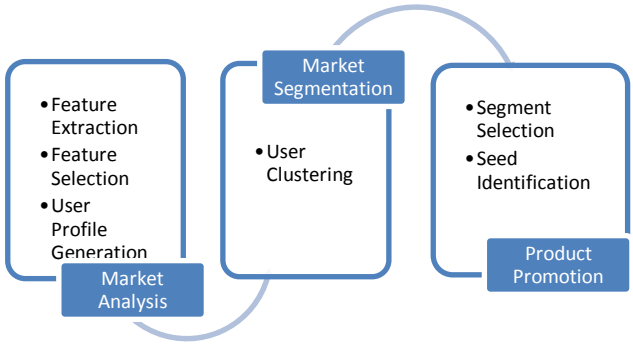
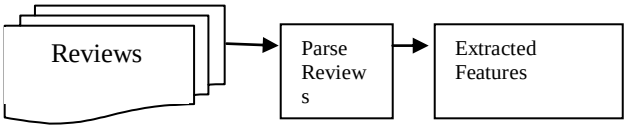


Figure 2: Three Phases of E-Marketing Model



Excellent picture quality, and the flexibility

```
<S>
<NG>
<WC="JJ">Excellent</W>
<WC="NN">picture</W>
<WC="NN">quality</W>
</NG>
<WC=",">,</W>
<WC="CC">and</W>
<NG>
<WC="DT">the</W>
<WC="NN">flexibility</
W>
</NG>
</S>
```

picture quality
flexibility

Figure 3: Feature Extraction

B. *Phase II: Market Segmentation*

This phase generates clusters of users as per their interest in similar product features. Clustering is a commonly used technique for customer segmentation and targeting. Clustering problem refers to grouping of customers in *k* segments such that each group contains customers similar to each other and, the difference between clusters is maximized. The simplest and the most popular clustering algorithm is *k-means* algorithm. It is very efficient, due to its linear time

complexity but the deterministic local search used in algorithm, may converge to the nearest local optima. Various meta heuristics and swarm intelligence based evolutionary algorithms [15]-[22] have been applied for clustering problem. However these algorithms applied heuristic and evolutionary approaches to avoid convergence to local optima with limited success rate. The viability of firefly algorithm for clustering was studied in [23]. It was observed that FClust, a firefly based algorithm for clustering has higher probability to achieve optimality.

C. Phase III: Targeted Product Promotion

This phase identifies the best segment(s) for the product to be promoted. The selection of segment and the number of segments considered for product promotion depends upon the type of product to be promoted and marketing budget. The key concept of this model is that it does not promote products to all or some random number of users of the segments. It focuses on limited number of initial users called *seeds* that have strong influence in the market this module exploits the social connectedness of users to identify the best seeds that encourage faster adoption of the product throughout the entire population. Since marketers provide complimentary or discounted products to these seeds so the decision regarding the number of seeds to be employed depends upon the marketing budget. Each individual who gets the awareness of the product is said to be *influenced*. Thus the main aim is to identify *k* initial users that maximize the profit within a given seeding budget. The problem of selecting optimal seeds that result in maximum influence in large network is denoted as *k-Max-Influence* problem and explained in following section.

III. k-Max Influence Problem

Given a social network G(V,E) and an integer k, find k seeds such that the incurred influence is maximized.[3]

A social network is modeled as a directed graph *G(V, E),* where V set of vertices represents users in the networks and E set of edges represents social interaction between users. Weight $w_{u,v}$ associated with each edge(u,v) indicates the probability of node *u* to influence node *v* in G. Nodes that are influenced by a product are called active and others are called inactive. Initially all nodes are inactive. The influence process starts with the set $S \subseteq V$ of nodes called seeds and activates them. These seeds in turn activate some of their neighbors (according to the information diffusion model). These newly active nodes then influence some of their neighbors, and so on. Hence the influence starts from the set S and cascades in the graph through the outgoing edges of the active nodes. The aim of the influence maximization problem is to choose the initial seed set S so that final influence (i.e. the number of active nodes at the end of the cascade) is maximized.

Domingos and Richardson introduced this problem to the field of computer science by posing the influence maximization as an optimization problem of selecting the best *k* seeds [2]. Further this problem had been studied under popular cascade models and the greedy approximation algorithm with a provable approximation guarantee (1-1/e- of the optimal solution) based on sub-modular property is proposed [3]. However this greedy algorithm significantly outperforms the classic degree and centrality-based heuristics in influence spread but suffers from high computation cost on estimating the influence function. Therefore "Cost-Effective Lazy Forward" (CELF) scheme for seed selection was proposed in [4]. The CELF optimization uses the submodularity property of the influence maximization objective to greatly reduce the number of evaluations on the influence spread of vertices. However, this improved algorithm still takes a few hours to complete in a graph with a few tens of thousands of vertices, so it is still not efficient for large-scale networks. Subsequently new heuristic scheme using a local arborescence structure presented in [1] proved to be the most efficient and scalable algorithm in their experiment. However identifying certain number of initial seeds from extremely large population is difficult. Therefore efforts had been made to define a local viral marketing problem (LVMP) which is opposed to global viral marketing problem (GVMP) [24]. The problem of GVMP is attacked by accessing the influencing probability between two users by analyzing their log of actions [25]. Other marketing researchers have explored how innovations diffuse across a variety of different topologies [10] and how word-of- mouth affects product adoption [5]. All these techniques did succeed in finding the best seeds but required global knowledge of the network i.e. knowledge about every node in the network and how it is connected to every other node which is unrealistic requirement in many real-world cases and time consuming. This paper tackles this problem by applying two promising evolutionary algorithms i.e. Differential evolution DE and firefly algorithm FA.

Differential Evolution (DE) has attracted much attention recently as an effective approach for solving numerical optimization problems. It is a stochastic population based algorithm developed by Storn and Price in 1995 [26]. It optimizes a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones Due to its simplicity it has been successfully applied in diverse fields of engineering [27]-[34]. Firefly algorithm (FA) is a recently introduced nature inspired approach for solving nonlinear optimization problem proposed by Yang in 2008[35],[36]. The algorithm is based on the behavior of social insects (fireflies) where each firefly has its own agenda and coordinates with other fireflies in the group (swarm) to achieve the same. This flashing behavior of fireflies is studied and incorporated in various techniques such as constrained continuous optimization tasks [11], feature selection [13], task graph scheduling [37], travelling salesman problem [38], job scheduling problem [39], supervised clustering problem [40] .

The two propagation model [3] used for information diffusion namely Independent Cascade Model (IC) [41] and Linear Threshold Model (LT) [42] are as follows:

A. Independent Cascade Model:

In this model initially all users are presumed to be inactive except for the users belongs to seeds set S. The diffusion process involves a number of steps. When a node *u* first becomes active at step *t*, it is given a single chance to activate

each currently inactive neighbor *v*. The activation of neighbor depends upon the $w_{u,v}$ associated with each directed edge(u,v). If node *u* succeeds, then node *v* will become active in step t+1; but irrespective of the success of *u* it cannot make further attempts to activate *v* in subsequent rounds. The process goes on until no more activation is possible.

B. Linear Threshold Model:

In this model a node *v* is influenced by each neighbor *u* according to weight $w_{u,v}$ such that

$$\sum_{u\ neighbor\ of\ v} w_{uv} \leq 1 \quad (1)$$

Now each node *v* chooses a threshold $\theta_v$uniformly at random from interval [0,1];this represents the weighted fraction of v's neighbor that must become active in order for *v to* become active. In order to activate node *v* the total weight of its active neighbor is at least θ_v. Our study concentrates only on IC mode but it can be extended to LT model.

$$\sum_{u\ active\ neighbor\ of\ v} w_{uv} \geq \theta_v \quad (2)$$

IV. Greedy Approach

Kempe proposed natural greedy strategy to tackle k-Max-Influence problem [3]. It takes the graph G and number *k* as input. The algorithm generates a seed set S of cardinality k, with the intention that the expected number of vertices influenced by the seed set S is maximum. The algorithm adds one vertex into the set S with each iteration *i* such that this vertex together with current set S maximizes the influence spread. Thus the vertex selected at iteration *i* is the one that maximizes the incremental influence spread during that iteration. Fig 4 describes the algorithmic steps of greedy approach. However this approach provides approximation guarantees arbitrarily close to (1-1/e) but takes too much time to calculate influence spread.

```
Void Greedy ( )
{
   initialize S= Ø, t=0
   while (t<=k)
    {
     for each vertex  v ∈ V\S
         {
          Sv=0
          calculate Influence spread Sv= {SU{v}}
          }
      S = SU {arg max_{v∈V\S} {s_v}}
       t=t+1
      }
}
```

Figure 4: Greedy algorithm for K-Max Influence

The working of greedy algorithm is illustrated with the help of graph shown in fig.2. The graph has six nodes (users) namely *a,b,c,d,e* and *f* respectively. Each edge (u,v) represent the social relationship between *u* and *v* and the weight w_{uv} associated with each edge indicates the probability of node *u* to influence node *v*. The main objective is to identify the set S ⊆ V of cardinality *k* where *k*=2 with the intention that the expected number of vertices influenced by the seed set S is maximum.

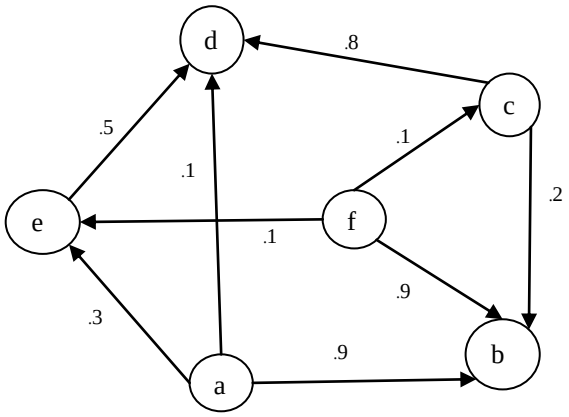


Figure 5: Example of Social Network

The algorithm first calculates the expected influencing value $E_{a,b,c,d,e,f}$ = {2.45,1,2.0,1,1.5,2.25}[1] of each node using diffusion model and probability information. Since node a has highest influencing value so it is selected as the first seed (S={a}). The second seed is selected from rest of the nodes i.e. {b,c,d,e,f} and the corresponding influences are {2.25, 2.0, 1.74, 1.38}. These values are conditional to '*a*' already being selected as a seed. Thus the second selection is {c} and final seed set is (S={a,c}) with the expected influence of 4.1.

V. Evolutionary Approach

The greedy algorithm significantly outperforms the classic degree and centrality-based heuristics in influence spread but suffers from high computation cost on estimating the influence function. This work explores the viability of evolutionary approach for k-Max Influence problem.

The key idea is to create a population of candidate solutions for an optimization problem, which is refined by alterations in the consecutive iteration. Candidate solutions are selected according to a fitness function, which evaluates their quality with respect to the optimization problem.

[1]We can compute the influence for a small graph by considering all cascades through a node e.g. Ea is the sum of influence on the nodes (a,b,d,e} = 1+.9+ (.1+(1-.1) *.3*.5)+.3≈2.45; where the first term is due to a's influence on itself while the third enumerates expectation on two possible paths from a to d.

Thus the paper compares two prominent evolutionary algorithms i.e. Differential Evolution (DE) and Firefly (FA) for their suitability to *k-Max-Influence* problem these algorithms are explained in detail as follows.

A. Differential Evolution (DE):

Differential evolution [26] is a numerical optimization approach which is very simple to implement with little or no parameter tuning. In this approach each individual j, chooses three other individuals *k*, *l* and *m* randomly from the population (with j≠k≠l≠m), then the difference of the chromosome *k* and *l* is calculated and scaled it by multiplying with parameter *f*.

This result is then added to the chromosome *m* to create an offspring. The approach does not create the entire chromosome of the offspring in this way but the genes are partially inherited from individual j using equation 3. Figure 6 represents the algorithmic steps of this evolutionary approach.

$$x.gene_i = \begin{cases} m.gene_i + f.(k.gene_i - l.gene_i)\ if\ \cup(0,1) < p \\ j.gene_i\ otherwise \end{cases} \quad (3)$$

```
void DE(G, k)
{
    initialize population(), t=0
    while(t<max_generations)
    {
        for each individual j
              {
               evaluate fitness f(j)
               create offspring x using eq 3
               evaluate f(x)
               if f(x)>f(j)
                  replace j with x
              }
    }
}
```

Figure 6: DE algorithm for K-Max Influence

B. Firefly Algorithm (FA):

Firefly algorithm (FA) [35] is inspired by biochemical and social aspects of real fireflies. Real fireflies produce a short and rhythmic flash that helps them in attracting their mating partners (regardless of their sex) and also serves as protective warning mechanism. FA formulates this flashing behavior with the objective function of the problem to be optimized. The movement of a firefly *i* attracted to another (brighter) firefly *j* is determined by equation 4.

$$x_i(t+1) = (1-\beta)x_i(t) + \beta x_j(t) + \mu_i \quad (4)$$

where β is the attractiveness of firefly *j*. It states how strongly it attracts firefly *i* in the swarm and is calculated using equation 5.

$$\beta = \beta_0\, e^{-\gamma r_{ij}} \quad (5)$$

where $r_{ij} = d(x_i, x_j)$, a Euclidean distance between two firefly *i* and *j*. In general $\beta_0 \in [0,1]$, describes the attractiveness at r=0 i.e. when two fireflies are found at the same point of search space S. The value of γ determines the variation of attractiveness with increasing distance from communicated firefly. It is basically the light absorption coefficient and generally lies in range of [0,10] [35],[36]. The randomized step μ_i moves between lower and upper bounds. Yang [35] proposed to use min $\mu_i = -0.5\,\alpha$ and max $\mu_i = 0.5\,\alpha$ with $\alpha \in [0,1]$ as in equation 6. Fig 7 depicts algorithm for FA.

$$\mu_i = \alpha(random{\sim}\mathbb{U}(0,1) - 0.5) \quad (6)$$

```
void FA()
{
        initialize population(), t=0
        while(t<=max_generations)
        {
          for each individual i
            {
              find most attractive  partner  j using eq. 5
               move i towards j in order to improve its
              fitness using eq 3
              evaluate new position of i i.e new f(i)
              if new f(i)> f(i)
                replace i with new i
              else
                move i using equation 6
            }
          t=t+1
        }
}
```

Figure7: FA algorithm for K-Max Influence

VI. EXPERIMENTAL STUDY

An experimental study to evaluate the performance of the above described algorithms was conducted. All the algorithms were executed in python 2.7 language and performed on a Pentium IV 400 MHz personal computer on the following datasets. The QUADRIVALENCY model is used to generate the weights for each edge that uniformly chooses a value from set {0.1, 0.25, 0.5, 0.75}. This probabilistic graph G is converted into G' by removing the edge (u, v) with the probability $1 - w_{u,v}$ to reduce the running time of algorithms as suggested in [1]. The influence spread of set (S) for G' is simply the set of vertices reachable from S in G' which can be obtained by linear scan (DFS or BFS).

The experiments were conducted to measure the total influence spread with given seed set *k* for the original graph and the sample graph (consists of few nodes of the original graph). Graph samples are randomly generated from the original graphs. The sampling process randomly selects the *t* number of nodes from the original set of nodes and constructs the edges by keeping the original edges between the selected nodes. For fair comparison all the algorithms were executed on that same sample with a given value of *k*. The pseudo-code for sampling approach is given in fig 8. This graph sampling is performed 100 times for each *t* value where *t* is the number of nodes in the graph. The following section describes experimental setup and the data set used for experiments.

```
void sampling (t)
 {
```

randomly select *t* number of nodes from the original set of nodes
compute sample graph G' by keeping the original edges between the selected nodes
for each edge (uv) in G'
{
w_{uv}= $random \sim \mathbb{U}(0.1,0.25,0.5,0.75)$
}
compute G'' by removing each edge from G' with probability 1-p
}

Figure 8 pseudo-code for graph sampling

A. Data Sets:

Five real data sets were used in the experimental study. Three data sets are collected from Stanford Large Network Dataset Collection [43] viz. Epinions, Wiki-Vote and Slashdot. Two data sets are collected from two different sections of the e-print arXiv5 [38] viz NetHEPT and NetPHY. The features of these data sets are summarized in table 1.

1) Wiki-Vote:

It is Wikipedia voting data from the inception of Wikipedia till January 2008. Nodes in the network represent Wikipedia users and a directed edge from node *i* to node *j* represents that user *i* voted on user *j*.[44].

2) Epinions:

It is who-trust-whom network at Epinions.com, where each node represents a member of the site and the link from member *u* to member *v* means that *u* trusts *v* (i.e. *v* has certain influence on *u*) [45].

3) Slashdot:

Slashdot contains friend links between the users of slashdot. Slashdot is a technology-related news website known for its specific user community. It allows users to tag each other as friends. [46].

4) NetHEPT:

It is a real-life academic collaboration networks from the "High Energy Physics - Theory" section with papers from 1991 to 2003. Each node in the network represents an author, and the number of edges between a pair of nodes is equal to the number of papers the two authors collaborated [38].

5) NetPHY:

It is a real-life academic collaboration networks from "Physics" section with papers from 1991 to 2003. Each node in the network represents an author, and the number of edges between a pair of nodes is equal to the number of papers the two authors collaborated [38].

Table1: Summarization of Data Set

Data Set	No. of Nodes	No. of Edges
Wiki-vote	7115	103689
Epinion	75888	508837
Slash	82168	948464
NetHEPT	15233	58891
NetPHY	37154	231584

B. Parameter Settings:

The following parameters are used to conduct experimental study.

1) Parameter k:

The variations in the performance of the algorithm with respect to maximum number of nodes influenced with seed set of size k is studied in this paper. The experiments for original graph were conducted on following values of k i.e. 10,20,30,40 and 50. The experiments for sample graph were conducted on two values of k i.e. 0.5% and 1% of the total nodes in the sample graph.

2) Sampling Parameters:

The graph sampling method given in fig 5 was applied on two different values of t (no. of nodes) i.e.2000 and 4000 for each data set. Table 2 depicts the details of the ten sample graphs generated from the above mentioned data sets. Thus the performance of each algorithm is evaluated on these varied types of samples.

Table 2: Details of samples generated.

Data Set	Cases	No. of Nodes	Averages No. of Edges
VOTE	A	2000	3300
	B	4000	10860
EPINION	C	2000	144
	D	4000	674
SLASH	E	2000	972
	F	4000	2402
NetHEPT	G	2000	551
	H	4000	2179
NetPHY	I	2000	500
	J	4000	2119

3) Algorithmic Parameters:

For both Firefly and DE the search process is started by randomly selecting 50 individuals that form the solution set from search space which is optimized during 200 generations. Each solution consists of a *k* number of users which is defined as seed size. The algorithms were executed for 10 times and the best result was used for comparisons. The parameter settings for DE and Firefly are depicted in table 3.

Table 3: Parameters for DE and FA.

DE		FA	
Parameter	**Value**	**Parameter**	**Value**
Population Size	50	Population Size	50

Crossover Rate	0.7	β_0	1
Factor	2	$\gamma.\alpha$	1

VII. Performance Analysis

The performance of the algorithms was measured with respect to total influence spread for a given value of *k* =. 10,20,30,40 and 50 for original network and k= 0.5% and 1% of total nodes for the sample cases mentioned in table 2. Table 4 to 8 show the results of algorithms for sample cases (A,B), (C,D), (E,F), (G,H) and (I,J) respectively. It is observed that firefly algorithm (FA) is able to identify the set S with maximum influence I(S) in all cases; in comparison to the sets identified by differential evolution (DE) and greedy algorithm. Moreover a higher gain is also observed in case of FA as compared to DE and greedy by increasing the value of k from 0.5% to 1% (i.e. from 10 to 20) for case A, C, E, G,I as shown in fig 9 and (i.e. from 20 to 40) for B,D,F,H,J as shown in fig 10. Figure 11 to 15 show the results for original network Vote, Epinion, Slash, NeTHEPT and NeTPHY respectively. The results reveal that both evolutionary approaches DE and FA perform well as in comparison to Greedy approach. However amongst the evolutionary approaches FA outperform DE for all the values of k in each case. Thus it reveals that both evolutionary approaches DE and FA perform well as in comparison to Greedy approach with respect to maximum influence incurred as well as marginal gain achieved by increasing the value of *k*. FA performs well for all the values of *k* and maintains the consistency in its results even with varied connectivity of nodes in graphs. Thus results show that FA algorithm has higher probability to identify the maximum influence within the given budget as compared to DE and Greedy.

Table 4: Maximum Influence for Case A and B.

	A		B	
	K=0.5%	K=1%	K=0.5%	K=1%
GREEDY	504	514	1188	1204
DE	507	517	1190	1208
FA	510	522	1199	1220

Table 5: Maximum Influence for Case C and D

	C		D	
	K=0.5%	K=1%	K=0.5%	K=1%
GREEDY	30	40	138	164
DE	33	43	189	217
FA	38	50	201	234

Table 6: Maximum Influence for Case E and F.

	E		F	
	K=0.5%	K=1%	K=0.5%	K=1%
GREEDY	47	57	316	336
DE	56	67	319	342
FA	61	74	324	350

Table 7: Maximum Influence for Case G and H.

	E		F	
	K=0.5%	K=1%	K=0.5%	K=1%
GREEDY	18	28	39	62
DE	23	36	52	79
FA	25	40	57	87

Table 8: Maximum Influence for Case I and J.

	E		F	
	K=0.5%	K=1%	K=0.5%	K=1%
GREEDY	17	26	41	63
DE	20	33	62	84
FA	24	36	69	96

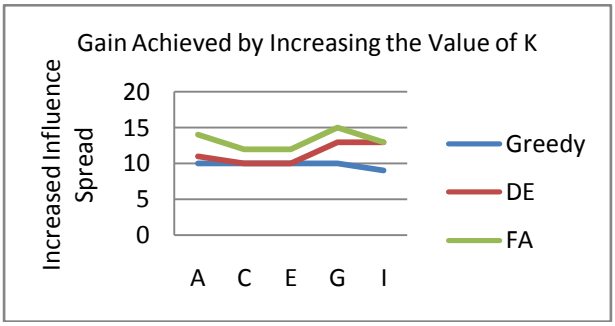


Figure 9: Gain by increasing the value of k from 10 to 20

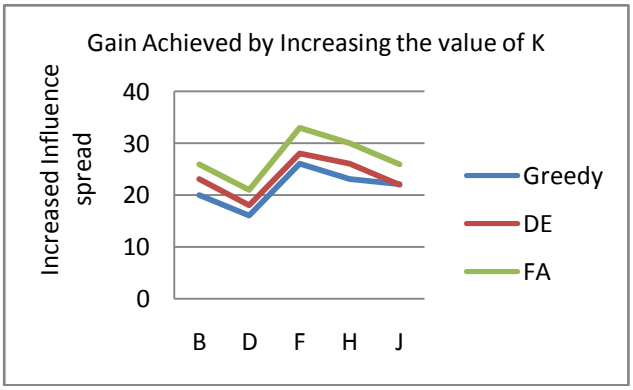


Figure 10: Gain by increasing the value of k from 20 to 40

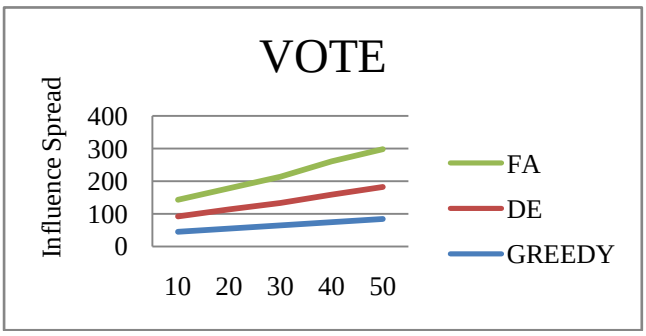


Figure 11: Influence Spread for Vote Data Set

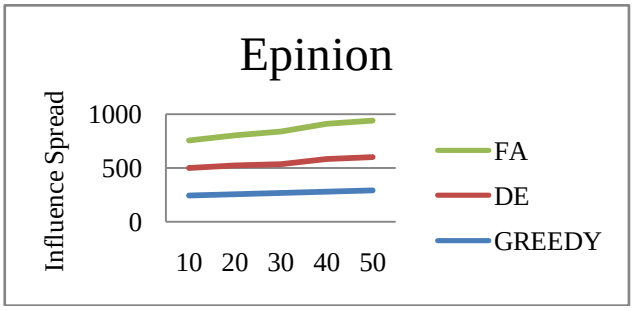


Figure 12: Influence Spread for Epinion Data Set

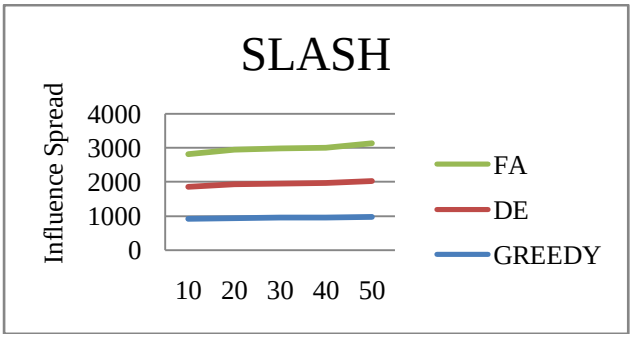


Figure 13: Influence Spread for Slash Data Set

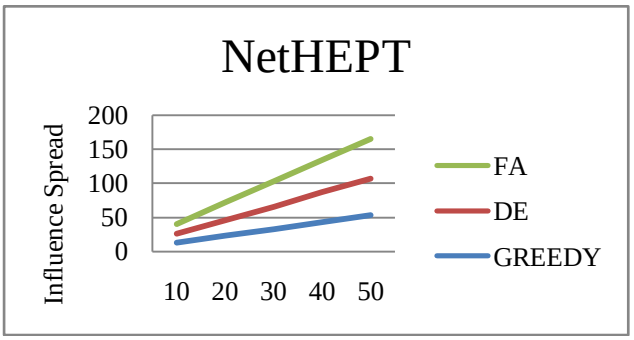


Figure 14: Influence Spread for NetHEPT

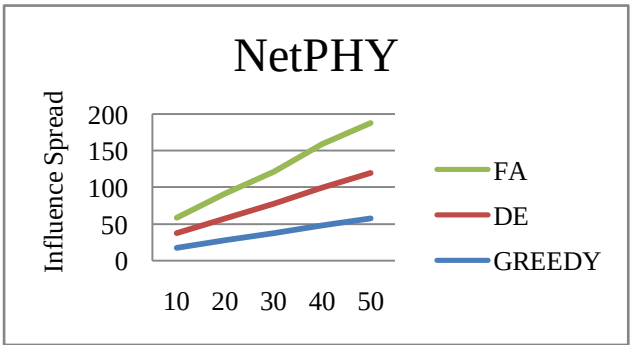


Figure 15: Influence Spread for NetPHY Data Set

VIII. Conclusion

E-marketing strategies that take advantage of influence propagation through social networks need to identify optimal seeds that results in maximum product awareness. Maximization of influence spread with the limited seeding budget (*k)* in large network is denoted as *k-Max-Influence* problem. Previous works tackled this problem by generalizing the greedy hill climbing techniques that suffer from high computation cost. The paper studied the viability of two promising evolutionary algorithms, Differential Evolution (DE) and Firefly algorithm (FA) for this problem. The experiments show promising results of employing evolutionary approach to solve k-max influence problem. Experimental study on Epinions, Wiki-Vote, Slashdot, NetHEPT and NetPHY datasets with respect to the total influence spread with given value of *k* and gain incurred by increasing the value of *k* was conducted. The results reveal that both evolutionary algorithms perform better as compared to greedy approach with respect to maximum influence incurred as well as gain achieved by increasing the value of *k*. However amongst the evolutionary algorithms FA maintains consistency in its results even with varied connectivity of nodes in graphs. Thus the performance of FA algorithm is superior and has higher probability to incur maximum influence spread within the given budget as compared to DE and Greedy.

References

[1] W. Chen, C. Wang, Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks", Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, July 25-28, 2010, Washington, DC, USA.

[2] P. Domingos and M. Richardson, "Mining knowledge-sharing sites for viral marketing", Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, July 23-26, 2002, Edmonton, Alberta, Canada.

[3] D. Kempe, J. Kleinberg and E. Tardos, "Maximizing the spread of influence through a social network", Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, August 24-27, 2003, Washington, D.C.

[4] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, "Cost-effective outbreak detection in networks", Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, August 12-15, 2007, San Jose, California, USA.

[5] J. Chevallier and D. Maylin, "The effect of word-of -mouth on sales: Online book reviews", Journal of Marketing Research 43,3(2006),345-354.

[6] H. Banati, M. Bajaj, "Promoting Products Online Using Firefly Algorithm", 12th International Conference on Intelligent System Design and Applications (ISDA-2012), November 27-29 2012, India, page 580-585.

[7] S. Datta, A. Majumder, and N. Shrivastava, "Viral marketing for multiple products," in *ICDM*, 2010.

[8] M. Kimura, and K. Saito, "Tractable models for information diffusion in social networks," *PKDD*, 2006.

[9] J. A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker Gordon, L. R. Riedl, : "GroupLens: Applying Collaborative Filtering to Usenet News", Communications of the ACM, 40(3):77–87, 1997.
[10] R. Sarker, M. Mohammadian, and X. Yao, eds., Evolutionary Optimization, vol. 48 of International Series in Operations Research and Management Science. Boston: Kluwer Academic, 2002.
[11] S. Lukasik, S. Zak, "Firefly algorithm for continuous constrained optimization tasks", ICCCI 2009, LNCS 5796, pp 97-106.
[12] H. Banati and M. Bajaj, "Dynamic User Profile Generation by Mining User Opinion", proceedings of 3rd international conference on computer engineering and technology (ICCET-11), June 17-19, 2011, Malaysia, Page 540.
[13] H. Banati and M.Bajaj, "Firefly based feature selection approach", Int. J. Computer Science Issues, vol. 8, Issue 4, No. 2, July 2011 473-480.
[14] H. Banati and M. Bajaj, "Feature Based Implicit User Modeling" 4th Indian International Conference on Artificial Intelligence (IICAI-09), December 16-18, 2009, India.
[15] K.S. Al-Sultan, Pattern Recognition,.28, 1443-1451. 1995
[16] S. Bandyopadhyay and U. Maulik, "Genetic clustering for automatic evolution of clusters and application to image classification", Pattern Recognition 2002, 35, 1197-1208.
[17] S. Bandyopadhyay, C.A.Murthy, and S.K.Pal, "Pattern classification with genetic algorithms", Pattern Recognition Letters,1995, 16, 801-808.
[18] S.Bandyopadhyay, C.A. Murthy, and S.K. Pal, "Theoretic performance of genetic pattern classifier", Journal of the Franklin Institute 1999, 336, 387-422.
[19] S. Bandyopadhyay, S. K. Pal, and C. A. Murthy C.A., "Simulated Annealing based pattern classification", Journal of Information Sciences, 1998, 109, 165-184.
[20] S. Paterlini , T. Krink , "Differential evolution and particle swarm optimization in partitional clustering", computational statistics and data analysis 2006, 50(5),1220-1247.
[21] S.Z. Selim, K.S. Al-Sultan, Pattern Recogn. 1991 24, 1003-1008.
[22] P.S. Shelokar, V. K. Jayaraman and B. D. Kulkarni, "An ant colony approach for clustering", Analytica Chimica Acta, 2004, vol. 509, pp. 187—195.
[23] H. Banati, M. Bajaj, "Performance Analysis of Firefly algorithm for Data Clustering. Accepted in International Journal of Swarm Intelligence.
[24] Forrest Stonedahl, William Rand, Uri Wilensky, Evolving viral marketing strategies, Proceedings of GECCO'10 Proceedings of the 12th annual conference on Genetic and evolutionary.
[25] A. Goyal, F. Bonchi, V.S. Laks Lakshmanan, "Learning influence probabilities in social networks", Proceedings of the third ACM international conference on Web search and data mining, February 04-06, 2010, New York, New York, USA.
[26] R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces", Technical Report TR-95-012, ICSI 1995.
[27] R. Storn, "Differential evolution design of an iir-filter", IEEE International Conference on Evolutionary Computation IEEE, 1996, pp. 268-273.
[28] S. Das and A. Konar, "Automatic image pixel clustering with an improved differential evolution Applied Soft Computing" 9, 2009 , no. 1, 226-236.
[29] R. Joshi and A.C. Sanderson, "Minimal representation multisensor fusion using differential evolution, IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans. 29, 1999, no. 1, 63-76.
[30] U. Maulik and I. Saha, Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery, Pattern Recognition 42, 2009, 2135-2149.
[31] T. Rogalsky, R. W. Derksen and S. Kocabiyik, "Differential evolution in aerodynamic optimization", Proc. of 46th Annual Conference of Canadian Aeronautics and Space Institute, 1999, pp. 29-36.
[32] R. Storn, "On the usage of differential evolution for function optimization", Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), IEEE, Berkeley, 1996, pp. 519-523.
[33] M. Varadarajan and K. S. Swarup, "Differential evolution approach for optimal reactive power dispatch", Applied Soft Computing 8 2008, no. 4, 1549-1561.
[34] M.K.Venu, R. Mallipeddi and P. N. Suganthan, "Fiber bragg grating sensor array interrogation using differential evolution", optoelectronics and Advanced Materials - Rapid Communications 2 ,2008, no. 11, 682-685.
[35] X.S. Yang, "Nature Inspired Metaheuristic Algorithms" 2008, Luniver Press, Beckington, UK..
[36] X.S.Yang, "Firefly algorithm for multimodal optimization", SAGA 2009, LNCS 5792, pp.169-178.
[37] U. Hönig, "A firefly algorithm-based approach for scheduling task graphs in homogenous systems", Proceeding Informatics, 2010 doi:10.2316/P.2010.724-033, 724.
[38] http://www.arXiv.org
[39] G. K. Jati and S. Suyanto, "Evolutionary discrete firefly algorithm for travelling salesman problem", ICAIS2011, Lecture Notes in Artificial Intelligence (LNAI 6943), pp.393-403 .
[40] J. Senthilnath, S.N. Omkar and V. Mani, "Clustering using firefly algorithm: Performance study", Swarm and Evolutionary Computation. 2011 doi:10.1016/j.swevo.2011.06.003.
[41] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of wordof- mouth," Marketing Letters, vol. 12(3), p. 211223, 2001.
[42] M. Granovetter, "Threshold models of collective behavior," *The American Journal of Sociology*, vol. 83, no. 6, pp. 1420– 1443, 1978.
[43] http://snap.stanford.edu/data.
[44] J. Leskovec, D. Huttenlocher and J. Kleinberg, "Predicting Positive and Negative Links in Online Social Networks". WWW 2010.
[45] M. Richardson, R. Agrawal, and P. Domingos, "Trust Management for the Semantic Web". ISWC, 2003.
[46] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney, "Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters". Internet Mathematics 6(1) 29--123, 2009.

Author Biographies

First Author Dr Hema Banati completed her Ph,d(2006) after her Masters in Computer Applications(M.C.A) both from Department of Computer Science, University of Delhi, India. At present she is an Associate Professor in the Department of Computer Science, Dyal Singh College, University of Delhi. She has over 18 years of teaching experience to both undergraduate and postgraduate classes. Over the past decade she has been pursuing research in the areas of Web engineering, software engineering, Human Computer Interaction, multiagent systems, E-commerce and E-learning. She has many national and international publications to her credit.

Second Author Ms. Monika Bajaj is a research scholar in Computer Science Department at University of Delhi. Her research interests include Web engineering, Human Computer Interfaces and E-commerce. She has many national and international publications to her credit.