

Research Article

Application of the Ant Colony Optimization Algorithm to the Influence-Maximization Problem

Wan-Shiou Yang and Shi-Xin Weng

Department of Information Management, National Changhua University of Education, #1, Jin-De Road, Changhua City 50007, Taiwan
Address correspondence to Wan-Shiou Yang, wsyang@cc.ncue.edu.tw

Received 27 March 2012; Accepted 28 May 2012

Abstract Consumers often form complex social networks based on a multitude of different relations and interactions. These interactions influence the decisions they make about adopting products or behaviors, and hence a company could receive a large cascade of further recommendations if it can identify and target influential consumers. This research borrowed from swarm intelligence—specifically the ant colony optimization algorithm—to address the influence-maximization problem. The proposed approaches were evaluated using a coauthorship data set from the arXiv e-print (www.arxiv.org), and the obtained experimental results demonstrated that our approaches outperform two well-known benchmark heuristics.

Keywords influence maximization; social network; ant colony optimization algorithm; e-commerce

1 Introduction

Consumers often form complex social networks based on a multitude of different relations and interactions. A piece of information can quickly spread between individuals within the social network in the form of “word-of-mouth” communication. A company can exploit such effects of social networks when marketing a new product. For example, if a company can identify and target a relatively small number of influential consumers, this could trigger a cascade of influence by which these people will recommend the new product to their friends.

The above influence-maximization problem has been formally defined by Domingos and Richardson [8] and Kempe et al. [11] as follows: given a social network and a prescribed number k , pick the k most “influential” individuals that will function as the initial adopters of a new product, so as to maximize the final number of infected individuals, subject to a specified model of influence diffusion. This influence-maximization problem has been extended as follows to the problem of introducing a new product into a market where competing products exist [5]: given the competitor’s choice of initial adopters

of technology B, maximize the spread of technology A by choosing a set of initial adopters so as to maximize the expected spread of technology A.

Several studies have addressed this influence-maximization problem. Kempe et al. [11] approached the problem from the perspective of two models of the diffusion of information: the threshold model and the cascade model. They showed that the underlying objective function of the problem is NP-hard and has monotonicity and submodularity properties. This led Kempe et al. [11,12] to apply a well-known greedy approximation to solve the original problem and competitive extensions thereof. Many of the existing approaches for solving the influence-maximization problem are based on approximation algorithms and assume that the objective function is monotonic and submodular [1,4,5,6,13,14].

However, as identified by Borodin et al. [3], there is a complex and broad family of diffusion models, and the properties of monotonicity and submodularity may not hold—in which case the greedy approach cannot be used. Therefore, in this study we borrowed from swarm intelligence—specifically the ant colony optimization (ACO) algorithm—to address the influence-maximization problem. The proposed approaches were evaluated using a coauthorship data set from the arXiv e-print (www.arxiv.org), and the obtained experimental results demonstrated that our approaches outperform two well-known benchmark heuristics.

This paper is organized as follows. Section 2 reviews related studies, and Section 3 describes the proposed approaches applying the ACO algorithm to the influence-maximization problem. The results of evaluating the proposed approaches are reported in Section 4, and Section 5 concludes with a summary of this study and a discussion of future research directions.

2 Literature review

A social network is a set of individuals connected through socially meaningful relationships, such as friendship, coworking, or information exchange [20]. Social networks

are formed when people interact with each other, and thus manifest in many aspects of everyday life. Social network theory traditionally views social relationships in terms of nodes and links [20], where the nodes are the individual actors within the networks and the links are the relationships between them. A social network plays a fundamental role as a medium for spreading information, ideas, or influence among its members [11,20]. These interactions influence decisions made by the individuals, and may allow any idea or innovation to make significant inroads into the population. Many diffusion models have been proposed for investigating how an idea or innovation spreads through a social network [3,17,19].

Motivated by applications to marketing, Domingos and Richardson [8] defined the influence-maximization problem as finding a k -node set that maximizes the expected number of influenced nodes at the end of the diffusion process. The authors modeled this problem using an arbitrary Markov random field, and provided heuristics for identifying individuals who exert a large overall effect on the network. Richardson and Domingos [16] extended their models to the continuous case so that businesses can allocate marketing funds more effectively. Kempe et al. [11] introduced various diffusion models such as the threshold model and the cascade model. They showed that determining an optimal seeding set is NP-hard, and that a natural greedy strategy yields provable approximation guarantees if the diffusion model has the properties of monotonicity and submodularity. This line of research was extended by introducing other competitors so as to produce the most far-ranging influence [1,4,5,6,12,13,14].

As noted by Borodin et al. [3], certain diffusion models—particularly those for investigating competitive influence in social networks—may not be monotonic or submodular, and hence the original greedy approach cannot be used. In the present study we therefore exploited the search capacity of the ACO algorithm to find an (approximate) solution for the influence-maximization problem. ACO, initially proposed by Dorigo [9], is a meta-heuristic developed for composing approximate solutions. ACO is inspired by observations of the collective foraging behavior in real ant colonies and represents problems as graphs, with solutions being constructed within a stochastic iterative process by adding solution components to partial solutions. Each individual ant constructs part of the solution using an artificial pheromone and heuristic information dependent on the problem. ACO has been receiving extensive attention due to its successful applications to many NP-hard combinatorial optimization problems [2,7,10,18].

3 Proposed approaches

The advent of recent information techniques, especially in communication systems (e.g., email, bulletin

boards, and messaging) and contacting systems [e.g., ICQ (www.icq.com), Friendster (www.friendster.com), and Facebook (www.facebook.com)], has led to a rapid growth in computer-mediated social networks. Data on the connectedness of humans can be obtained by links either explicitly stated by consumers (e.g., friendships stated on contacting systems) or implicitly inferred from previous interaction data (e.g., email logs). In this research we transform the connectedness data into a consumer's social network and represent the network as a directed graph, where each node represents a consumer and each edge represents the connectedness between two consumers. Our view is formally described in Definition 1.

Definition 1. A social network SN is a directed graph $SN = (V, E)$, where V is the set of nodes and E is the set of edges. Each node $V_i \in V$ represents a consumer, and each edge (V_i, V_j) , where $V_i, V_j \in V$, represents the connectedness from V_i to V_j .

We assume that a company has a fixed budget for targeting k consumers who will trigger a cascade of influence. If S is the initial set of adopters, then its influence is expressed as the expected number of adopters at the end of the diffusion process (see Definition 2).

Definition 2. Given a set S of initial adopters, the influence of S , denoted as $\text{Inf}(S)$, is the expected number of the adopters at the end of the diffusion process, subject to a specified diffusion model ς .

In the case that there is no competing product, our goal is to identify a set of initial adopters S of size k that will maximize the expected number of the adopters influenced by S (see Definition 3).

Definition 3 (without a competing product). Given a social network $SN = (V, E)$, a prescribed number k , and a diffusion model ς , our goal is to find a set of nodes S , $S \subseteq V$ and $|S| = k$, that maximizes $\text{Inf}(S)$.

In the case that there is a competing product, we consider the influence-maximization problem from the follower's perspective. Therefore, suppose the competitor's choice of initial adopters is C , our goal is to choose a set of initial adopters that will maximize the expected spread of our new product (see Definition 4).

Definition 4 (with a competing product). Given a social network $SN = (V, E)$, a set C of initial adopters of a competing product, a prescribed number k , and a diffusion model ς , our goal is to find a set of nodes S , $S \subseteq V - C$ and $|S| = k$, that maximizes $\text{Inf}(S)$.

The inspiration for ACO is the foraging behavior of real ants [10]. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as

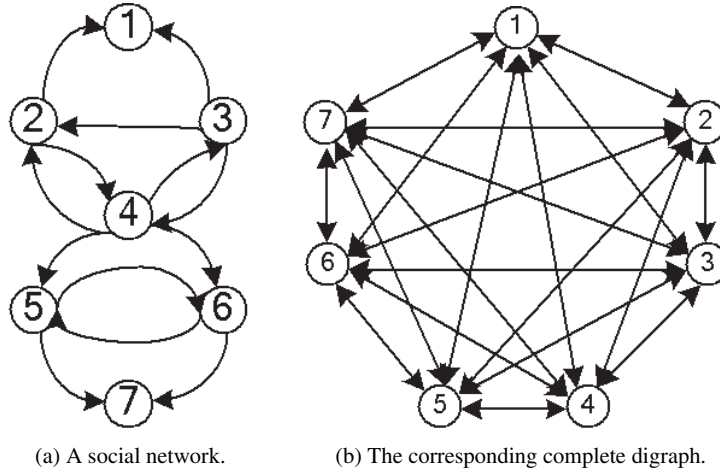


Figure 1: An example of a social network and its corresponding complete digraph.

an ant finds a food source, it evaluates the quantity and quality of the food and carries some of it back to the nest. During the return trip, the ant deposits a chemical pheromone trail on the ground. The quantity of pheromone deposited depends on the quantity and quality of the food, and this will guide other ants to the food source. Indirect communication between the ants via pheromone trails enables them to find the shortest paths between their nest and food sources. This characteristic of real ant colonies is exploited in artificial ant colonies, and the ACO algorithm utilizes a graph representation to find (approximate) solutions for the target problem. We construct a complete digraph to represent the original social network (see Definition 5).

Definition 5. A complete digraph $SN' = (V, E')$ is constructed to represent the original social network, $SN = (V, E)$. Each node in SN' represents a node in SN , and for each pair of nodes (V_i, V_j) in SN , a bidirectional edge (V_i, V_j) is constructed in SN' .

Figure 1(a) shows an example of a social network of size 7, and Figure 1(b) shows the corresponding complete digraph.

Also, we transform the defined influence-maximization problem into a problem of finding a circle of prescribed length so as to maximize the expected spread from the set of nodes in the circle (see Definitions 6 and 7).

Definition 6 (without a competing product). Given a social network $SN = (V, E)$, a corresponding complete digraph $SN' = (V, E')$, a prescribed number k , and a diffusion model ς , our goal is to find a circle S , $S \subseteq V$ and $|S| = k$, on SN' that maximizes $\text{Inf}(S)$ on SN .

Definition 7 (with a competing product). Given a social network $SN = (V, E)$, a corresponding complete digraph $SN' = (V, E')$, a set C of initial adopters of a competing product, a prescribed number k , and a diffusion model ς ,

```

ACO_InfluenceMaximization()
{
    InitializePheromoneValue();
    While (termination conditions not met)
    {
        GenerateSolutions();
        EvaluateSolutions();
        UpdatePheromoneValue();
    }
    Return best solution;
}

```

Figure 2: The basic ACO algorithm for the influence-maximization problem.

our goal is to find a circle S , $S \subseteq V - C$ and $|S| = k$, on SN' that maximizes $\text{Inf}(S)$ on SN .

The central component of an ACO algorithm is a parameterized probabilistic model, which is called the pheromone model. This model is used to probabilistically generate solutions to the problem under consideration by assembling them using a finite set of solution components. At run-time, ACO algorithms update the pheromone values using previously generated solutions. The update aims to concentrate the search within regions of the search space containing high-quality solutions. We therefore design a basic ACO algorithm as shown in Figure 2, which works as follows. The algorithm first initializes all of the pheromone values according to the `InitializePheromoneValue()` function. An iterative process then starts, with the `GenerateSolutions()`

InitializePheromoneValue(SN' : a complete digraph; k : integer; m : integer)

```

{
  For each node  $i$  in  $SN'$ 
  {
     $\tau(j) = \varepsilon$ , // pheromone on  $i$ 
     $l = 1$ ; solution  $Si = \emptyset$ ;
    While  $|Si| < k$ 
    {
       $Si = Si \cup$  nodes randomly selected from the level- $l$  neighbors of node  $i$ ;
       $l++$ ;
    }
  }
  Sort all solutions by their influence value  $Inf()$ ;
  Let Top be the set of top- $m$  solutions;
  For each solution  $Si$  in Top
  {
    For each node  $j$  in  $Si$ 
    {
       $\tau(j) = \tau(j) + Inf(Si)$ ;
    }
  }
}

```

Figure 3: The InitializePheromoneValue() function.

function being used by all ants to probabilistically construct solutions to the problem based on a given pheromone model in each iteration. The EvaluateSolutions() function is used to evaluate the quality of the constructed solutions, and some of the solutions are used by the UpdatePheromoneValue() function to update the pheromone before the next iteration starts.

The InitializePheromoneValue() function is used to initialize the pheromone values of all nodes of the constructed complete digraph. Initially, each node has a very small pheromone value of $\varepsilon \neq 0$. A possible solution is then created for each node by assembling the solution components as follows. Starting node i is added first, and each of its first-level neighbors are independently selected with probability p ; then its second-level neighbors are selected, and so on, until k nodes are assembled in the solution. The influence of the solution—which corresponds to the expected number of the adopters at the end of the diffusion process—is then evaluated. The influences of the top- m solutions are then used as the pheromone and lay down on all component nodes of the solution. Different solutions may lay down pheromone values on the same nodes, in which case all pheromone values of the same node are summarized. The detailed algorithm is shown in Figure 3.

Table 1: Example of pheromone values.

Solution no.	Nodes	Influence
1	1,2,6	4
2	2,1,4	5
3	3,1,2	4
4	4,3,5	6
5	5,6,7	3
6	6,5,7	3
7	7,6,2	4

Consider the example shown in Figure 1(b). Suppose each solution has three nodes and that each node has an initial pheromone value of 1. The InitializePheromoneValue() function creates 7 solutions since there are 7 nodes in the complete digraph. Suppose each solution is created and the influence of each solution is evaluated as listed in Table 1. Then nodes 3, 4, and 5 will have a pheromone value of 7 and the other nodes all have a pheromone value of 1 if only the best solution (i.e., solution 4) lay down its pheromone.

In the iterative process, all ants probabilistically construct solutions to the problem. In the GenerateSolutions() function, each artificial ant generates a complete target set by choosing the nodes according to a probabilistic state-transition rule: an ant positioned on node r chooses node s to move to by applying the rule given by

$$s = \begin{cases} \arg \max_{u \in V-r} \{ [\tau(u)]^\alpha * [\eta(u)]^\beta \}, & \text{if } q \leq q_0 \\ S, & \text{otherwise,} \end{cases} \quad (1)$$

where q is a random number uniformly distributed in $[0, 1]$, q_0 is a parameter ($0 \leq q_0 \leq 1$), τ is the pheromone value, η is the heuristic value, and S is a random variable selected according to the probability distribution given by

$$p(u) = \frac{[\tau(u)]^\alpha * [\eta(u)]^\beta}{\sum [\tau(u)]^\alpha * [\eta(u)]^\beta}. \quad (2)$$

The above state-transition rule clearly favors transitions toward nodes with large pheromone and heuristic values. Parameter q_0 determines the relative importance of exploitation versus exploration: every time an ant in node r has to choose a node s to move to, it samples a random number $0 \leq q \leq 1$. If $q \leq q_0$, the best node (according to (1)) is chosen (exploitation); otherwise a node is chosen according to (2) (biased exploration).

We also propose using three methods for determining the heuristic values of nodes:

- (1) Degree centrality approach: degree centrality is defined as the number of links incident upon a node [15]. Since outdegree is often interpreted as a form of gregariousness in a social network [15], we define the number of

GenerateSolutions(k : integer, q_0 : integer, m : integer): a set of solutions

```

{
  Repeat
  {
     $S_i = \emptyset$ ,  $n=1$ ;
    While ( $n \leq k$ )
    {
       $q$ =a random number  $\in [0, 1]$ ;
      If  $q < q_0$ 
      {
         $S_i = S_i \cup$  the node with the maximum  $[\tau(u)]^\alpha * [\eta(u)]^\beta$  value;
      }
      Else
      {
        Randomly select a node  $a$  according to probability distribution  $\frac{[\tau(u)]^\alpha * [\eta(u)]^\beta}{\sum [\tau(u)]^\alpha * [\eta(u)]^\beta}$ ;
         $S_i = S_i \cup a$ ;
      }
       $n++$ ;
    }
    Until  $m$  solutions are generated;
  }
}

```

Figure 4: Algorithm of the GenerateSolutions() function.

links that connects the node to other nodes as its degree heuristic value. For the example shown in Figure 1(a), the degree heuristic value of node 4 is 4.

- (2) Distance centrality approach: distance centrality is another commonly used influence measure [15]. The distance centrality of a node is defined as the average distance from this node to all of the other nodes in the graph. Again considering node 4 in Figure 1(a), its distance centrality is 1.33 since its distances from nodes 1, 2, 3, 5, 6, and 7 are 2, 1, 1, 1, 1, and 2, respectively. We define the distance heuristic value of a node as the number of nodes minus its distance centrality.
- (3) Simulated influence approach: for each node i , we also use its influence $\text{Inf}(i)$ as its heuristic value. However, as described by Kempe et al. [11], it is an open question to compute this quantity exactly. We therefore simulate the random process to obtain a feasible estimate. Specifically, given a particular diffusion model, we simulate the process N times, and compute the average number of influenced nodes for each node as its heuristic value.

The detailed algorithm of the GenerateSolutions() function is shown in Figure 4.

First suppose the pheromone and heuristic values of all nodes in Figure 1 are updated as listed in Table 2. Then suppose that an artificial ant is going to choose a 3-node solution, and that three random numbers are generated: 0.6, 0.9, and 0.5. Let $\alpha = 1$, $\beta = 1$, and $q_0 = 0.8$. For the first node, the ant will select node 4 since this has the largest

Table 2: Example pheromone values and the heuristic values of nodes.

Node	1	2	3	4	5	6	7
Pheromone value	1	1	7	7	7	1	1
Heuristic value	0	2	3	4	2	2	0

EvaluateSolutions(S : a set of nodes; ς : a diffusion model): a number

```

{
  Set parameters of diffusion model  $\varsigma$ ;
   $Sum=0$ ;
  For  $i=1$  to 1000
  {
     $n$ =the number of nodes influenced by target set  $S$ ;
     $Sum+=n$ ;
  }
  Return  $Sum/1000$ ; // return expected number of influenced nodes
}

```

Figure 5: Algorithm of the EvaluateSolutions() function.

value according to (1); for the second node, since $q_0 < 0.9$ the ant will select one node according to the probability distribution given in (2); suppose that node 6 is selected in this step. Finally, the ant will select node 3 since this node has the largest value among the leaving nodes. A set of nodes $\{4, 6, 3\}$ is then generated as the solution.

The EvaluateSolutions() function is then used to evaluate the performance of each solution. The performance of a target set S is evaluated by computing the value of $\text{Inf}(S)$. Again, we obtain estimates by simulating the diffusion models in a random process. Specifically, given a particular diffusion model, we simulate the process N times, and compute the average number of influenced nodes for each target set. A detailed algorithm is shown in Figure 5.

Once all ants have found their target sets, the pheromone is updated on all nodes. In our system the global updating rule is implemented according to

$$\tau(u) = (1 - \rho) * \tau(u) + \sum_{k=1}^m \tau(u)^k. \quad (3)$$

Similar to the InitializePheromoneValue() function, the influences of the top- m solutions are used as the pheromone and lay down on all component nodes of the solution, and all pheromone values of the same node are summarized. The ρ parameter is the evaporation rate and is implemented to avoid the algorithm converging too rapidly toward a suboptimal region. The detailed algorithm of the UpdatePheromoneValue() function is shown in Figure 6.

```

UpdatePheromoneValue()
{
    For each solution  $Si$  in Top
    {
        For each node  $u$  in  $Si$ 
        {
             $\tau(u) = (1-\rho) * \tau(u) + Inf(Si);$ 
        }
    }
}

```

Figure 6: Algorithm of the UpdatePheromoneValue() function.

Consider the example in Table 2. Let ρ be 0.9. Suppose there is an artificial ant who finds a 3-node solution $\{3, 4, 6\}$, whose expected influence $Inf()$ is 5, and the current pheromone values of nodes 3, 4, and 6 are 7, 7, and 1, respectively. After updating the pheromone, these values will be set as 11.3, 11.3, and 5.9, respectively.

The iterative process of the ACO_InfluenceMaximization() function ends when some termination condition is met, such as exceeding the execution time limit or a certain ratio of the nodes being influenced. The result, which is the best target set, is then returned.

4 Evaluation

We evaluated the efficacy of the proposed approaches by conducting experiments on a real world coauthorship data set. The coauthorship network was compiled from the complete list of papers on the arXiv e-print dated between January 1, 2006 and December 31, 2010. We constructed a coauthorship network as a directed graph in which each node represents an author and each directed edge represents a coauthor relationship from the author to another author (i.e., if they have coauthored at least one paper). Each edge (s_i, s_j) in the constructed coauthorship network is associated with a weight defined as $\frac{|A_i \cap A_j|}{|A_i|}$, where A_i and A_j denote the sets of papers authored by s_i and s_j , respectively. The coauthorship network contained 8,436 nodes representing all of the authors of the included papers,

Table 3: Best combinations of parameters α , β , and ρ in the noncompetitive case.

Heuristic:	Degree heuristic	Distance heuristic	Simulated influence heuristic
Parameters α - β - ρ	1-1-0.8	1-1-0.8	1-2-0.9

and 168,712 edges representing the coauthor relationships between these authors.

We first evaluated the proposed approaches in the non-competitive case. In this case we used the original linear threshold model [3] as the diffusion model. In this model, each node v initially chooses a threshold $\theta_v \in [0, 1]$ that represents the minimum fraction of active neighbors necessary for the activation of v . Also, each directed edge (u, v) is assigned a weight $w_{u,v} \in [0, 1]$ that roughly characterizes the weight of the influence that u has over v . The diffusion process unfolds in discrete steps, during which new node v becomes activated if the total weight of the incoming edges from its activated neighbors, A_v , is larger than its threshold (i.e., $\sum_{u \in A_v} w_{u,v} \geq \theta_v$).

The first set of experiments aimed at finding the best combinations of the α , β , and ρ ACO parameters of the proposed approaches in the noncompetitive case. These experiments used α values of 0, 1, 2, 3, and 4; β values of 0, 1, 2, 3, and 4; and ρ values of 1, 0.9, 0.8, 0.7, and 0.6. Each combination of different α , β , and ρ values corresponded to a single experiment. Also, the size of a target set and random number q_0 were set as 30 and 0.8, respectively. We simulate the diffusion process N times for each targeted set and compute the averaged influence. Previous runs indicate that the quality of approximation after 1000 iterations is comparable to that after 10000 or more iterations. In this and subsequent experiments, we therefore simulate the diffusion process 1000 times. Table 3 lists the best combinations of parameters α , β , and ρ of the proposed approaches with different heuristics.

The results in Table 3 indicate that the performance of the proposed approaches was best with $\alpha = 1$, $\beta = 1$, and $\rho = 0.8$ for the degree centrality heuristic; $\alpha = 1$, $\beta = 1$, and $\rho = 0.8$ for the distance centrality heuristic; and $\alpha = 1$, $\beta = 2$, and $\rho = 0.9$ for the simulated influence heuristic. These settings were therefore used in the subsequent experiment. Also, it is worth noting that α and β are not equal to 0 in these best settings, which indicates that both the pheromone and heuristic values contribute to the performance of the proposed approaches.

We then compared the performances of the proposed approaches in the noncompetitive case. This experiment used two benchmarks—the maximum degree approach and the minimum distance approach—as baselines for our comparisons. In the maximum degree approach, we simply pick k nodes in the coauthorship network having the k

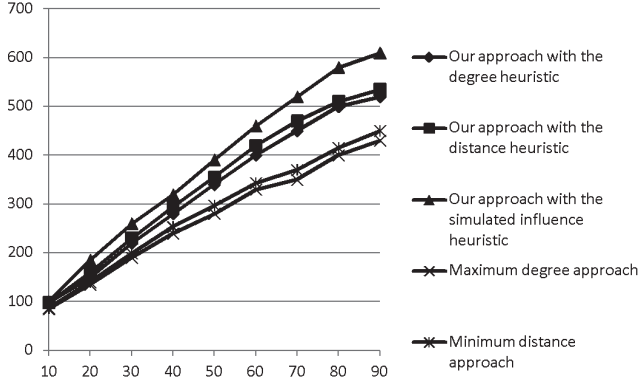


Figure 7: Comparison of the performances of the proposed approaches and benchmarks in the noncompetitive case

highest degree centrality values. In the minimum distance centrality approach, we pick k nodes in the coauthorship network having the k lowest distance centrality values. For our approach the three different heuristics described in Section 3 were used. These values were averaged over 1000 runs. Figure 7 shows the average spread of the approximate solutions generated by our approaches and two benchmarks when solution size k was 10, 20, 30, 40, 50, 60, 70, 80, and 90.

It can be seen that the performance was best for our approach with the simulated influence heuristic, followed by our approach with the distance heuristic, our approach with the degree heuristic, the minimum distance approach, and the maximum degree approach (in that order). The performances of all of our three proposed approaches were better than those of the two benchmarks. The experimental results demonstrate the effectiveness of the search capacity of the ACO algorithm. Also, for the three proposed approaches, the approach using the simulated influence heuristic had the highest diffusion values. This indicates that the simulated influence heuristic was superior to the degree and distance heuristics.

We also compared the performances of the proposed approaches in the competitive environment. In this case we used the weight-proportional competitive linear threshold model [3] as the diffusion model. In this model, each node v initially chooses a threshold $\theta_v \in [0, 1]$, and each directed edge (u, v) is assigned a weight $w_{u,v} \in [0, 1]$. Given sets I_A and I_B of initial adopters, the diffusion process unfolds as follows. In each step t , every inactive node v checks the set of edges incoming from its active neighbors. If their collective weight exceeds the threshold value, the node becomes active. In that case the node will adopt technology A with a probability equal to the ratio between the collective weight of edges outgoing from A active neighbors and the total collective weight of edges outgoing from all active neighbors. It has been proven that the weight-proportional

Table 4: Best combinations of parameters α , β , and ρ in the competitive case.

Heuristic:	Degree heuristic	Distance heuristic	Simulated influence heuristic
Parameters α - β - ρ	1-2-0.9	1-2-0.8	1-3-0.9

competitive linear threshold model does not have the properties of monotonicity and submodularity [3].

We again attempted to find the best combinations of parameters α , β , and ρ of the proposed approaches in the competitive case. These experiments investigated the same values of α , β , and ρ as used in the noncompetitive case. At the beginning of each experiment, a set of 30 nodes was randomly selected as the initial adopter of a competitive product. Also, the size of a target set and random number q_0 were again set as 30 and 0.8, respectively. Table 4 lists the best combinations of parameters α , β , and ρ of the proposed approaches with different heuristics.

The results in Table 4 indicate that the performance of the proposed approaches was best with $\alpha = 1$, $\beta = 2$, and $\rho = 0.9$ for the degree heuristic; $\alpha = 1$, $\beta = 2$, and $\rho = 0.8$ for the distance heuristic; and $\alpha = 1$, $\beta = 3$, and $\rho = 0.9$ for the simulated influence heuristic. These settings were used in the subsequent experiment. Also, comparison with the experimental results in Table 3 reveals that β was larger in the competitive case. It is inferred that using heuristics to select influential nodes plays a more important role in the competitive case.

We finally compared the performances of the proposed approaches in the competitive case. This experiment also used the maximum degree approach and minimum distance approach as baselines for our comparisons. Moreover, the size of a target set and random number q_0 were again set as 30 and 0.8, respectively. At the beginning of the experiment, a set of 30 nodes was randomly selected as the initial adopter of a competitive product. Figure 8 shows the average spread of the approximate solution generated by our approaches and two benchmarks when solution size k was 10, 20, 30, 40, 50, 60, 70, 80, and 90.

Figure 8 shows there were only small increases in the influenced nodes in the initial stage (i.e., $k = 10$ and $k = 20$), which is expected since the number of initial adopters of a competitive product (i.e., 30) is larger than k and hence more nodes may be influenced by the adopters of the competitive product. The diffusion values for the three proposed approaches were highest for the simulated influence heuristic. The performances of all of our three proposed approaches were again better than those of the two benchmarks. The experimental results show that the proposed approaches can be used even when the diffusion model does not have the properties of monotonicity and submodularity, and provides superior performance.

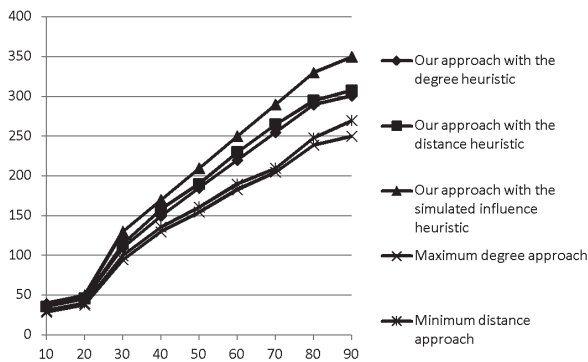


Figure 8: Comparison of the performances of the proposed approaches and benchmarks in the competitive case

5 Conclusions

This research used the search capacity of the ACO algorithm to solve the influence-maximization problem in both noncompetitive and competitive cases. The proposed approaches use the degree centrality, distance centrality, and simulated influence methods for determining the heuristic values. Experiments revealed that the proposed approach with the simulated influence heuristic provides the best performance.

Our work could be extended in several directions, such as testing the proposed methods in different social networks and using different diffusion models. It would also be interesting to investigate other heuristic methods that could further improve the proposed approaches.

References

- [1] S. Bharathi, D. Kempe, and M. Salek, *Competitive influence maximization in social networks*, in Proc. of the 3rd International Conference on Internet and Network Economics (WINE'07), San Diego, CA, 2007, 306–311.
- [2] C. Blum, *Ant colony optimization: introduction and recent trends*, Physics of Life Reviews, 2 (2005), 353–373.
- [3] A. Borodin, Y. Filmus, and J. Oren, *Threshold models for competitive influence in social networks*, in Proc. of the 6th International Conference on Internet and Network Economics (WINE'10), Stanford, CA, 2010, 539–550.
- [4] T. Cao, X. Wu, S. Wang, and X. Hu, *Maximizing influence spread in modular social networks by optimal resource allocation*, Expert Systems with Applications, 38 (2011), 13128–13135.
- [5] T. Carnes, C. Nagarajan, S. M. Wild, and A. van Zuylen, *Maximizing influence in a competitive social network: a follower's perspective*, in Proc. of the 9th International Conference on Electronic Commerce (ICEC '07), New York, 2007, 351–360.
- [6] W. Chen, Y. Wang, and S. Yang, *Efficient influence maximization in social networks*, in Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09), Paris, France, 2009, 199–208.
- [7] O. Cordon, F. Herrera, and T. Stützle, *A review on the ant colony optimization metaheuristic: basis, models and new trends*, Mathware & Soft Computing, 9 (2002), 141–175.
- [8] P. Domingos and M. Richardson, *Mining the network value of customers*, in Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01), San Francisco, CA, 2001, 57–66.
- [9] M. Dorigo, *Optimization, learning and natural algorithms*, PhD thesis, Politecnico di Milano, Milan, Italy, 1992.
- [10] M. Dorigo and C. Blum, *Ant colony optimization theory: a survey*, Theoretical Computer Science, 344 (2005), 243–278.
- [11] D. Kempe, J. Kleinberg, and E. Tardos, *Maximizing the spread of influence through a social network*, in Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03), Washington, DC, 2003, 137–146.
- [12] D. Kempe, J. Kleinberg, and E. Tardos, *Influential nodes in a diffusion model for social networks*, in Proc. of the 32nd International Conference on Automata, Languages and Programming (ICALP'05), Lisboa, Portugal, 2005, 1127–1138.
- [13] J. Kostka, Y. A. Oswald, and R. Wattenhofer, *Word of mouth: rumor dissemination in social networks*, in Structural Information and Communication Complexity, vol. 5058 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2008, 185–196.
- [14] E. Mossel and S. Roch, *Submodularity of influence in social networks: from local to global*, SIAM Journal on Computing, 39 (2010), 2176–2188.
- [15] T. Opsahl, F. Agneessens, and J. Skvoretz, *Node centrality in weighted networks: generalizing degree and shortest paths*, Social Networks, 32 (2010), 245–251.
- [16] M. Richardson and P. Domingos, *Mining knowledge-sharing sites for viral marketing*, in Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02), Edmonton, Canada, 2002, 61–70.
- [17] E. M. Roger, *Diffusion of Innovations*, Free Press, New York, 5th ed., 2003.
- [18] K. Thangavel, M. Karnan, P. Jeganathan, A. P. Lakshmi, R. Sivakumar, and G. Geetharamani, *Ant colony algorithms in diverse combinational optimization problems – a survey*, ACSE Journal, 6 (2006), 7–26.
- [19] T. Valente, *Network Models of the Diffusion of Innovations*, Hampton Press, New York, 1995.
- [20] S. Wasserman, K. Faust, D. Iacobucci, and M. Granovetter, *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK, 1994.