Application of the Ant Colony Optimization Algorithm to Competitive Viral Marketing

Wan-Shiou Yang and Shi-Xin Weng

Department of Information Management, National Changhua University of Education,
No. 1, Jin-De Rd., Changhua 50007, Taiwan
{wsyang,sxweng}@cc.ncue.edu.tw

Abstract. Consumers often form complex social networks based on a multitude of different relations and interactions. By virtue of these interactions, they influence each other's decisions in adopting products or behaviors. Therefore, it is essential for companies to identify influential consumers to target, in the hopes that influencing them will lead to a large cascade of further recommendations. Several studies, based on approximation algorithms and assume that the objective function is monotonic and submodular, have been addressed this issue of viral marketing. However, there is a complex and broad family of diffusion models in competitive environment, and the properties of monotonic and submodular may not be upheld. Therefore, in this research, we borrowed from swarm intelligence-specifically the ant colony optimization algorithm-to address the competitive influence-maximization problem. The proposed approaches were evaluated using a coauthorship data set from the arXiv e-print (http://www.arxiv.org), and the obtained experimental results demonstrated that our approaches outperform two well-known benchmark heuristics.

Keywords: Ant Colony Optimization Algorithm, Viral Marketing, Social Network, E-Commerce.

1 Introduction

Consumers often form complex social networks based on a multitude of different relations and interactions. By virtue of these interactions, they influence each other's decisions in adopting products or behaviors. Therefore, it is essential for companies to identify influential consumers to target, in the hopes that influencing them will lead to a large cascade of further recommendations. This influence-maximization problem can be defined as the following: Given a social network, 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.

Several studies have been addressed this influence-maximization problem. Kempe et al. [8] showed that the problem is NP-hard and many underlying diffusion models have monotonicity and submodularity properties. Hence, they applied a well-known greedy

I. Maglogiannis, V. Plagianakos, and I. Vlahavas (Eds.): SETN 2012, LNAI 7297, pp. 1–8, 2012.

approximation to solve the problem. 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] [3] [5] [10] [11].

This influence-maximization problem has been extended to introduce a new product into a market where competing products exist [4]: Given the competitor's choice of initial adopters of technology B, maximize the spread of technology A by choosing a set of initial adopters such that the expected spread of technology A will be maximal. As identified by Borodin et al. [2], however, there is a complex and broad family of competitive diffusion models, and the properties of monotonic and submodular may not be upheld-in which case the greedy approach cannot be used.

Therefore, in this research, we borrowed from the swarm intelligence-specifically the ant colony optimization (ACO) algorithm-to address the competitive influence-maximization problem. Our proposed approaches do not use the properties of monotonicity and submodularity and hence are general approaches. The proposed approaches were evaluated using a coauthorship data set from the arXiv e-print (http://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 competitive 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

Motivated by applications to marketing, Domingos and Richardson [6] defined the original influence-maximization problem as finding a *k*-node set that maximizes the expected number of convinced nodes at the end of the diffusion process. In [13], authors further extended their models to the continuous case. In [8], the authors introduced various diffusion models. 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] [3] [4] [5] [9] [10] [11].

As noted by Borodin et al. [2], however, certain diffusion models-particular those for investigating competitive influence in social networks-may not be monotonic or submodular, and hence the original greedy approach cannot be used. In this research, we exploited the search capacity of the ACO algorithm to find an (approximated) solution for the competitive influence-maximization problem. ACO, initially proposed by Dorigo [7], is a new meta-heuristic developed for composing approximated solutions. ACO is inspired by 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 a 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 today [7].

3 Proposed Approaches

In this research, we transform consumer's connectedness data into a social network and represent the network as a directed graph, where each node represents a consumer and each edge represents the connectedness between two nodes. In this research, we assume that a company has a fixed budget for targeting k consumers who will trigger a cascade of influence. We consider the competitive influence-maximization problem from the follower's perspective. Therefore, given a social network $SN=(V, E)$ and a set C of initial adopters of a competing product, our goal is to choose a set of nodes S, $S \subseteq V-C$ and $|S|=k$, that maximizes the spread of our new product.

The inspiration for ACO is the foraging behavior of real ants [7]. When searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates the quantity and the 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 (approximated) solutions for the target problem.

In order to utilize graph representation to find (approximated) solutions, in this research, we construct a complete digraph to represent the original social network. Then, we transform the defined competitive 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.

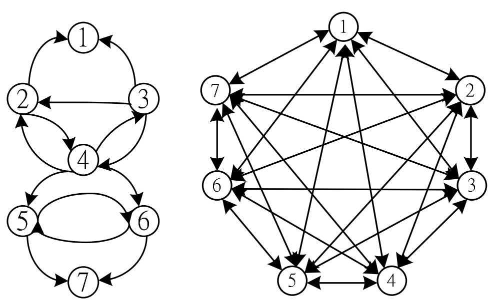
The central component of an ACO algorithm is a parameterized probabilistic model, which is called the pheromone model. This model is used to probabilistically generate solutions to the problem under consideration by assembling them using a finite set of solution components. At run-time, ACO algorithms 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 1, 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 GenerateSolution() functionbeing used by all ants to probabilistically construct solutions to the problem based on a given pheromone model in each iteration. The EvaluateSolution() 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.

```
ACO_InfluenceMaximization()
{
    InitializePheromoneValue();
    While (termination conditions not met)
    {
        GenerateSolution();
        EvaluateSolution();
        UpdatePheromoneValue();
    }
    Return best solution;
}
```

Fig. 1. The basic ACO algorithm for the competitive influence-maximization problem

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.

Figure 2(a) shows an example of a social network of size 7, and Figure 2(b) shows the corresponding complete digraph. Suppose each solution has 3 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.



(a) A social network (b) The corresponding complete digraph

Fig. 2. A social network and its corresponding complete digraph

Table 1. An example of pheromone values

Solution #	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

Then, in the iterative process, all ants probabilistically construct solutions to the problem. In the GenerateSolution() 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 the node *s* to move to by applying the rule given by (1). In (1), *q* is a random number uniformly distributed in [0,1], q_0 is a parameter ($0 \le q_0 \le 1$), and *S* is a random variable selected according to the probability distribution given in (2). In both (1) and (2), τ is the pheromone value and η is the heuristic value respectively.

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

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

Also, in this research, we propose using two methods for determining the heuristic values of nodes:

(a) Degree centrality approach: Degree centrality is defined as the number of links incident upon a node [12]. Since outdegree is often interpreted as a form of gregariousness in a social network, we define the number of links that the node directs to others as its degree heuristic value. For the example shown in Figure 2(a), the degree heuristic of node 4 is 4.

(b) Distance centrality approach: Distance centrality is another commonly used influence measure [12]. 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 2(a), its distance centrality is 1.33 since its distances from nodes 1, 2, 3, 5, 6, 7 are 2, 1, 1, 1, 1, and 2, respectively. We define the distance heuristic value of a node as the number of all nodes minus its distance centrality.

Suppose the pheromone and the heuristic values of all nodes in Figure 2 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 α =1, β=1, and q_0 = 0.8. For the first node, the ant will select node 4 since this has the largest value according to (1); for the second node, since 0.9 > q_0, 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 be generated as the solution.

Table 2. An example of the 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

The EvaluateSolution() function is then used to evaluate the performance of each solution. To evaluate the performance of a solution, we need to compute the expect spread of the solution. Again, we obtain estimates by simulating the diffusion models in a random process. Specifically, given a particular diffusion model, we simulate the process 1000 times, and compute the average number of influenced nodes for each solution.

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 (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 parameter ρ is the evaporation rate and is implemented to avoid the algorithm converging too rapidly toward a suboptimal region.

$$\tau(u) = (1-\rho) * \tau(u) + \sum_{k=1}^{M} \tau(u)^k \tag{3}$$

Considering 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 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 (= 7 * 0.9 + 5), 11.3 (= 7 * 0.9 + 5), and 5.9 (= 1 * 0.9 + 5) 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

In this section, 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 (www.arxiv.org) 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 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 and 168,712 edges representing the co-author relationships between these authors.

We compared the performances of the proposed approaches in the competitive environment. We used the weight-proportional competitive linear threshold model [2] 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 the 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 values, the node becomes active. In that case, the node will adopt technology *A* with probability equal to the ratio between the collective weight of edges outgoing from *A*-active neighbors and the total collective weight of edges out going from all active neighbors. It has been proven that the competitive model does not have the properties of monotonicity and submodularity [2]. We conducted several preliminary experiments to determine the ACO's parameters α, β, ρ for the proposed approaches. The best combination of parameters α-β-ρ is 1-2-0.8, and therefore this setting was used in the subsequent experiment.

We compared the performances of the proposed approaches in the competitive case. In this experiment, two benchmarks-the maximum degree approach and the minimum distance approach-were used as baselines for our comparisons. In the maximum degree approach, we simply pick *k* nodes in the coauthorship network having the *k* 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 two different heuristics described in Section 3 were used. These values were averaged over 1000 runs. Figure 3 shows the averaged spread of the approximated solution generated by our approaches and two benchmarks when solution size *k* was 10, 20, 30, 40, 50, 60, 70, 80, and 90.

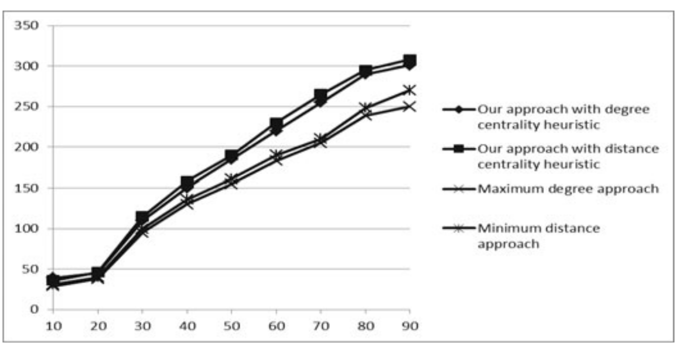


Fig. 3. Comparison of the performances of the proposed approaches and benchmarks in competitive case

It can be seen that our approach using distance heuristic value has the best performance, followed by our approach using degree heuristic value, the minimum distance centrality approach, and the maximum degree approach in order. The performances of both of our two 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, in the two proposed approaches, the approach using distance heuristic value has the highest diffusion values. It indicates that the distance centrality heuristic superior than the degree centrality heuristic.

5 Conclusions

This research used the search capacity of the ACO algorithm to solve the competitive influence-maximization problem. Experiments revealed that the proposed approach using distance heuristic value resulted in best performance. Our work could be extended in several directions, such as testing the proposed approaches in different social networks and using different diffusion models.

References

1. Bharathi, S., Kempe, D., Salek, M.: Competitive Influence Maximization in Social Networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 306–311. Springer, Heidelberg (2007)
2. Borodin, A., Filmus, Y., Oren, J.: Threshold Models for Competitive Influence in Social Networks. In: Saberi, A. (ed.) WINE 2010. LNCS, vol. 6484, pp. 539–550. Springer, Heidelberg (2010)
3. Cao, T., Wu, X., Wang, S., Hu, X.: Maximizing Influence Spread in Modular Social Networks by Optimal Resource Allocation. Expert Systems with Applications 38(10), 13128–13135 (2011)
4. Carnes, T., Nagarajan, C., Wild, S.M., Zuylen, A.: Maximizing Influence in a Competitive Social Network: A Follower's Perspective. In: International Conference on Electronic Commerce (2007)
5. Chen, W., Wang, Y., Yang, S.: Efficient Influence Maximization in Social Networks. In: International Conference on Knowledge Discovery and Data Mining (2009)
6. Domingos, P., Richardson, M.: Mining the Network Value of Customers. In: International Conference on Knowledge Discovery and Data Mining (2001)
7. Dorigo, M., Blum, C.: Ant Colony Optimization Theory: A Survey. Theoretical Computer Science 344 (2005)
8. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the Spread of Influence through a Social Network. In: International Conference on Knowledge Discovery and Data Mining (2003)
9. Kempe, D., Kleinberg, J., Tardos, É.: Influential Nodes in a Diffusion Model for Social Networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
10. Kostka, J., Oswald, Y.A., Wattenhofer, R.: Word of Mouth: Rumor Dissemination in Social Networks. In: Shvartsman, A.A., Felber, P. (eds.) SIROCCO 2008. LNCS, vol. 5058, pp. 185–196. Springer, Heidelberg (2008)
11. Mossel, E., Roch, S.: Submodularity of Influence in Social Networks: From Local to Global. SIAM Journal on Computing 39(6), 2176–2188 (2010)
12. Opsahl, T., Agneessens, F., Skvoretz, J.: Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths. Social Networks 32(245), 245–251 (2010)
13. Richardson, M., Domingos, P.: Mining Knowledge-sharing Sites for Viral Marketing. In: International Conference on Knowledge Discovery and Data Mining (2002)