MINIMIZING THE COMPLETE INFLUENCE TIME
IN A SOCIAL NETWORK WITH STOCHASTIC COSTS
FOR INFLUENCING NODES

YAODONG NI* and QIAONI SHI
School of Information Technology and Management
University of International Business and Economics, Beijing 100029, China

Received 14 July 2012
Revised 28 October 2012

In this paper, we study the problem of targeting a set of individuals to trigger a cascade of influence in a social network such that the influence diffuses to all individuals with the minimum time, given that the cost of initially influencing each individual is with randomness and that the budget available is limited. We adopt the incremental chance model to characterize the diffusion of influence, and propose three stochastic programming models that correspond to three different decision criteria respectively. A modified greedy algorithm is presented to solve the proposed models, which can flexibly trade off between solution performance and computational complexity. Experiments are performed on random graphs, by which we show that the algorithm we present is effective.

Keywords: Social networks; complete influence time; stochastic costs; stochastic programming; greedy algorithm.

1. Introduction

A social network is a graphical model of the relationships within a group of individuals, in which each node represents an individual and each edge denotes the relationship between the two individuals it connects. Social networks have attracted much attention from various areas, such as computer science,6,15,20 economics,1 sociology,23 epidemiology.5

The diffusion of influence in social networks plays an important role in many real applications.17,18 For example, in virus marketing, a new product is promoted to only a small number of people, but can be adopted by a large proportion of the population. This is because the recommendation of the product from one friend to another is actually a type of influence that spreads throughout the social network. A variety of models characterizing how influence diffuses have been proposed, such as the linear threshold model,10 the independent cascade model,9 the voter
model and the incremental chance model. Motivated by practical demands, decision makers attempt to optimize the target set of individuals to whom the influence is initially sent. A number of optimization problems with different objectives have been investigated, including maximizing the total number of influenced individuals,\textsuperscript{10,11} maximizing the expected lift in profit,\textsuperscript{6,21} minimizing the size of the initial target set of individuals that guarantees complete influence.\textsuperscript{3} A recently proposed problem is the complete influence time (CIT) minimization problem,\textsuperscript{17,18} which is to determine the set of the individuals who are targeted initially such that the time taken to influence all the individuals in the network is minimized.

The restriction of limited resources almost always exists in real world. For example, in the context of this paper, the budget for influencing the target set of individuals is usually limited. In a number of papers on influence diffusion in social networks,\textsuperscript{10,17} the restriction of limited resources is embedded in the assumption that the target set has a fixed size. In other works,\textsuperscript{7,18} a more general scenario is considered, where a predetermined amount of budget is given and the cost for initially influencing each individual may be different.

In real life, uncertainty always exists. In the problem we consider in the paper, the cost to initially influence an individual may be uncertain, because it usually depends on many dynamic or vague factors such as how the mood of the individual is when she gets influenced, to what extent the individual favors the influence, and so on. Taking these uncertain factors into account, it would be inappropriate to quantify the costs as constants. However, none of the above works discusses uncertain costs of influencing the target set of individuals.

In this paper, we focus on the CIT minimization problem where a limited amount of budget is provided and the cost for initially influencing each individual is given as a random variable. We adopt the incremental chance model to describe the diffusion of influence, because compared to that under other models, the diffusion process is progressive under the incremental chance model, which guarantees the complete influence that is required by the CIT minimization problem.

Since random variables are involved in the optimization problem we study, we consider three different decision criteria in the theory of stochastic programming, and propose three stochastic programming models. In order to solve the stochastic programming models, we present a modified greedy algorithm, which flexibly trades off between solution performance and computational complexity. A number of experiments are performed, by which we show that the modified greedy algorithm is effective.

The rest of this paper is organized as follows. We briefly introduce some preliminaries of the incremental chance model in Sec. 2. In Sec. 3, we present three stochastic programming models. Section 4 describes a modified greedy algorithm for solving the models. We perform experiments in Sec. 5 to show the effectiveness of the proposed algorithm and draw conclusions in Sec. 6.
2. The Incremental Chance Model

We denote a social network by a weighted undirected graph $G = (N, E, W, \varepsilon)$. Each node in $G$ represents one individual in the group we consider, and $N$ denotes the set of all nodes in $G$. Each edge in $G$ represents a relationship between the pair of individuals corresponding to the two nodes the edge connects, and the set of all edges is denoted by $E$. Since $G$ is an undirected graph, for any two nodes $i$ and $j$ in $N$, the edge connecting them, if there exists one, can be denoted by either $(i, j)$ or $(j, i)$. In the following of this paper, we call that $i$ and $j$ are neighbors if there exists an edge $(i, j)$. We denote by $W$ the weight function which assigns a positive weight to each edge to measure the strength of the relationship the edge represents. For example, a large weight $W(i, j)$ on edge $(i, j)$ means that $i$ and $j$ are closely related.

The cost for initially influencing a node $i$ in $G$ is assumed as a random variable $\varepsilon_i$. We let $\varepsilon = \{\varepsilon_i|i \in N\}$ be a random vector recording the stochastic costs for initially influencing all the individuals. The target set of individuals who are chosen to be influenced initially is denoted by a binary vector $x = \{x_i|i \in N\}$, where $x_i = 1$ if node $i$ is in the target set or $x_i = 0$ otherwise. Therefore, the cost for initially influencing the target set $x$ is

$$C(x, \varepsilon) = \sum_{i \in N} \varepsilon_i x_i.$$

The incremental chance model was proposed to characterize the diffusion of dominant influence. According to the incremental chance model, at any time step, a node is called an active node if it has been influenced, or otherwise called an inactive node. At any time step $t$, an inactive node $j$ turns to be active with the probability

$$p^j_t = \frac{\sum_{i \in N^a(j)} W(i,j)}{\sum_{i \in N(j)} W(i,j)},$$

where $N(j)$ denotes the set of neighbors of node $j$, and $N^a(j)$ is the set of active neighbors of $j$ at time step $t$. The probability $p^j_t$ is called the influence chance of $j$ at time $t$. Although the process of the diffusion of influence is stochastic, it is clear to describe. At the beginning, the nodes in the target set are influenced and set to be active, which triggers the cascade of influence. Then inactive nodes get influenced according to their influence chance. Note that the process is progressive, that is, active nodes will stay active and never turn to be inactive. Thus, the diffusion of influence will always terminate as all nodes in the social network will turn to be active ultimately with probability $1$, which achieves the complete influence. The duration of the process of influence diffusion is called the complete influence time (CIT). We denote the CIT by $\tau(x)$ given that the target set of nodes is $x$. 
3. The Stochastic Programming Models

In this paper, we focus on the following problem. Given a social network \( G \) and a predetermined amount of budget \( B \), how to determine the target set of individuals \( x \), to whom we initially send the influence, such that the cost for initially influencing \( x \) is under the budget and the CIT \( \tau(x) \) is minimized. The solution to the problem is not straightforward, because for a given \( x \), both \( C(x, \varepsilon) \), the cost for initially influencing \( x \), and the CIT \( \tau(x) \) are random variables, which makes the problem a stochastic programming problem. According to different decision criteria of decision makers, solutions to the problem may be quite different. In this section, we adopt three well-known decision criteria which are widely used to solve uncertain programming problems in various areas,\(^{12,13,16,19,22,24,25,26,27}\) and provide three stochastic programming models.

The first model is motivated by the natural idea that we may estimate a random variable by its expected value, which is commonly called the expected value model (EVM).\(^{14}\) In the EVM model for our problem, the objective is to minimize the expected value of the CIT \( E[\tau(x)] \), and a target set \( x \) is feasible if the expected value of the cost \( E[C(x, \varepsilon)] \) dose not exceed the budget \( B \). The EVM model is as follows:

\[
\begin{align*}
\min & \ E[\tau(x)] \\
\text{subject to:} & \ E[C(x, \varepsilon)] \leq B, \\
& x = \{x_i = 0 \text{ or } 1 \mid i \in N\}.
\end{align*}
\]

For many cases, estimating a random variable by its expected value is not always appropriate, especially for random variables with large variances. For our problem, when the variance of \( C(x, \varepsilon) \) is large, it is quite possible that \( C(x, \varepsilon) \) exceeds \( B \) with a high probability even though \( E[C(x, \varepsilon)] \leq B \) holds. Similarly, with a large probability, the CIT \( \tau(x) \) can be much greater than its expected value \( E[\tau(x)] \). In practice, decision makers often predetermine a confidence level \( \alpha \) as a safety margin and solve problems in the sense that the constraints are satisfied at least \( \alpha \) of time. This idea is usually referred to as chance-constrained programming (CCP).\(^2\) For our problem, the CCP model is as follows:

\[
\begin{align*}
\min & \ \{\tau \mid \Pr\{\tau(x) \leq \tau\} \geq \alpha\} \\
\text{subject to:} & \ \Pr\{C(x, \varepsilon) \leq B\} \geq \beta, \\
& x = \{x_i = 0 \text{ or } 1 \mid i \in N\},
\end{align*}
\]

where \( \alpha \) and \( \beta \) are two predetermined values between 0 and 1.

For some cases, decision makers may take a third decision criterion, under which they attempt to maximize the chance to meet some stochastic events that they expect to happen. In order to make decisions under this criterion, Liu provided the
dependent-chance programming (DCP). In our problem, there may be a deadline for the diffusion of influence such that exceeding the deadline will cause a huge penalty. In this view, we may need to maximize the probability of the event that the CIT does not exceed the deadline \( \tau_0 \). The DCP model is as follows:

\[
\begin{align*}
\max & \quad \Pr\{\tau(x) \leq \tau_0\} \\
\text{subject to :} & \\
E[C(x, \varepsilon)] & \leq B, \\
x & = \{x_i = 0 \text{ or } 1 \mid i \in N\},
\end{align*}
\]

where \( \tau_0 \) is a predetermined value representing the deadline.

4. A Modified Greedy Algorithm

It is usually difficult to get the optimal solution of a CIT minimization problem. One reason is that the analytic representations of the uncertain functions in the three models we propose are difficult to obtain. In this view, we employ the stochastic simulation techniques to estimate the values of the uncertain functions, including \( E[\tau(x)] \), \( E[C(x, \varepsilon)] \), \( \min \{\tau \mid \Pr\{\tau(x) \leq \tau\} \geq \alpha\} \), and \( \Pr\{\tau(x) \leq \tau_0\} \). Another reason why the problem is difficult is that a real social network usually contains a huge number of individuals, which makes searching the optimal solution very time-consuming. An efficient approach to solving optimization problems in complex systems is the greedy algorithm, which usually avoids the brute-force search and obtains satisfactory solutions. For the CIT minimization problems in social networks, by the traditional greedy algorithm, we put nodes in the target set once a time unless the constraints are violated. Every time we add a new node to the target set, we select the one that brings the most improvement in the value of the objective function. Whenever a node is put in the target set, it will never be taken out of the set, and thus the target set is efficiently constructed. However, traditional greedy algorithm computes the values of the objective functions in the proposed models by stochastic simulation, which can be still time-consuming for large-scale social networks.

In this paper, we modified the traditional greedy algorithm for the sake of saving the computation in stochastic simulation. The idea is to introduce an integer parameter \( r \) to trade off between solution performance and computational complexity. In the traditional greedy algorithm, when we need to add a new node to the target set, we compute, by stochastic simulation, the change of the value of objective function for every node that is possible to be selected. However, in the modified greedy algorithm, every time we add a node to the target set, we first determine \( r \) nodes, each of which is potentially a good choice for being put in the target set; and then compute by stochastic simulation the change of the value of objective function for adding each of the \( r \) nodes. We design several efficiently-computable heuristic functions to indicate which node is a potentially good alternative for being
put in the target set, thus the computational complexity of stochastic simulation is alleviated.

In order to present the heuristic functions, we first define the distance graph of a social network. The distance graph of a social network is a directed graph that has exactly the same set of nodes with the original social network. For each pair of nodes that are connected by an edge in the social network, there are two directed edges in opposite directions connecting them in the distance graph. The weight on each edge \((i, j)\) in the distance graph represents the distance between \(i\) and \(j\), which is defined by

\[
d(i, j) = \frac{\sum_{l \in N(j)} W(l, j)}{W(i, j)}.
\]

For each set of nodes \(x\) and a node \(i\), the heuristic function returns a non-negative value \(h(x, i)\) quantifying the fitness of putting \(i\) in \(x\) in terms of improving the performance of the solution. For example, if \(h(x, i) \geq h(x, j)\), then adding \(i\) to \(x\) is potentially better than putting \(j\) in \(x\). Here, we present three heuristic functions.

We define by \(SP(i, j)\) the length of the shortest path from \(i\) to \(j\) in the distance graph and by \(SP(x, i) = \min_{l \in x} SP(l, i)\) the shortest path length from set of nodes \(x\) to \(i\) in the distance graph. If \(x\) is an empty set, we define \(SP(x, i)\) as a sufficiently large number, for example, the sum of the weights on all the edges in the distance graph. The first heuristic function is motivated by the intuition that it usually takes much time to spread the influence to a node that is distant from the influenced nodes. We call the heuristic function “Shortest Path Length” (SPL) and define it by

\[
h_1(x, i) = SP(x, i).
\]

We define the maximin path length of a set of nodes \(x\) by \(\max_{j \in N} SP(x, j)\), which describes how far from the set to the farthest node in the network. The idea of the second heuristic function is that it gives a high score to a node if adding it to the target set will highly reduce the maximin path length. We call this heuristic function “Maximin Path Length Reduction” (MPLR) and define it by

\[
h_2(x, i) = \max_{j \in N} SP(x, j) - \max_{j \in N} SP(x \cup \{i\}, j).
\]

Given the distance graph of a social network, it is not difficult to generate a spanning forest with the set of roots being \(x\), such that each node in the forest is connected to its closest root. Such a spanning forest is called the shortest path forest and denoted by \(F_{SP}(x)\). Let \(\text{Edge}(F_{SP}(x))\) be the set of edges in \(F_{SP}(x)\). We present the third heuristic function named “Shortest Path Forest Weight Reduction” (SPFWR) as follows.

\[
h_3(x, i) = \sum_{(l, j) \in \text{Edge}(F_{SP}(x))} d(l, j) - \sum_{(l, j) \in \text{Edge}(F_{SP}(x \cup \{i\}))} d(l, j).
\]

Finally, we present the modified greedy algorithm for solving the three stochastic programming models in Table 1. For solving the EVM model (1), \(T(x) = \{i \in \ldots\)
\[ N \setminus x \mid E[C(x \cup \{i\}, \varepsilon)] > B \] and \[ D(x, i) = E[\tau(x)] - E[\tau(x \cup \{i\})] \]. For solving the CCP model (2), \[ T(x) = \{ i \in N \setminus x \mid \Pr \{ C(x \cup \{i\}, \varepsilon) \leq B \} < \beta \} \] and \[ D(x, i) = \min \{ \tau \mid \Pr \{ \tau(x \cup \{i\}) \leq \tau \} \geq \alpha \} - \min \{ \tau \mid \Pr \{ \tau(x) \leq \tau \} \geq \alpha \}. \] For solving the DCP model (3), \[ T(x) = \{ i \in N \setminus x \mid E[C(x \cup \{i\}, \varepsilon)] > B \} \] and \[ D(x, i) = \Pr \{ \tau(x \cup \{i\}) \leq \tau_0 \} - \Pr \{ \tau(x) \leq \tau_0 \}. \]

Table 1. The modified greedy algorithm for solving the stochastic programming models.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x = \emptyset )</td>
<td>( x = \emptyset );</td>
</tr>
<tr>
<td>while</td>
<td>while (</td>
</tr>
<tr>
<td>( n = 0 )</td>
<td>( n = 0; )</td>
</tr>
<tr>
<td>( R = \emptyset )</td>
<td>( R = \emptyset; )</td>
</tr>
<tr>
<td>while</td>
<td>while ( n &lt; r ) and (</td>
</tr>
<tr>
<td>( i_n = \arg \max_{i \in N \setminus (x \cup T(x) \cup R)} h(x, i) )</td>
<td>( i_n = \arg \max_{i \in N \setminus (x \cup T(x) \cup R)} h(x, i); )</td>
</tr>
<tr>
<td>( R = R \cup i_n )</td>
<td>( R = R \cup i_n; )</td>
</tr>
<tr>
<td>( n = n + 1 )</td>
<td>( n = n + 1; )</td>
</tr>
<tr>
<td>( j = \arg \max_{i \in R} D(x, i) )</td>
<td>( j = \arg \max_{i \in R} D(x, i); )</td>
</tr>
<tr>
<td>( x = x \cup {j} )</td>
<td>( x = x \cup {j}; )</td>
</tr>
<tr>
<td>return ( x )</td>
<td>return ( x );</td>
</tr>
</tbody>
</table>

5. Experiments

In this section, a number of experiments are performed to show the effectiveness of the modified greedy algorithm and how it trades off between solution performance and computational complexity. In all the following experiments, we set the number of nodes \( |N| = 1000 \) and the number of samples in the stochastic simulation \( M = 500 \). The experiments are conducted on a Pentium4 PC.

We assume the social networks as randomly-generated graphs. In order to generate a social network, we define a parameter \( p \) to specify how social the individuals in the social network will be. Given the set of nodes \( N \), for every pair of nodes in the network, we generate an edge between them with probability \( p \). Therefore, a high value of \( p \) will result in a social network where each individual is possibly connected to a lot of other individuals. After we generate the social network in such a stochastic way, we add as few edges as possible to make the network a connected graph. In the following experiments, without loss of generality, we randomly assign an integer weight from \( \{1, 2, 3\} \) to each edge. The stochastic cost for initially influencing each node \( i \) is given as follows: we first generate an integer \( u_i \) in the interval \([0, 20]\) at uniformly random, and then set the stochastic cost \( \varepsilon_i \) as the random variable with uniform distribution \( U(20, 20 + u_i) \).

In order to show the effectiveness of the proposed algorithm, in the first series of experiments, we solve the three stochastic programming models with different budget \( B \) by the modified greedy algorithm. In addition, we compare the solution performance of the modified greedy algorithm with that of a trivial heuristic method called “randomly choosing nodes”. The heuristic “randomly choosing nodes”
randomly selects the nodes in the target set without exploiting any information of the network. In the first series of experiments, without loss of generality, we set $p = 0.02$ to generate the social network and the trade-off parameter $r = 50$ in the modified greedy algorithm. The solutions for solving EVM model, CCP model and DCP model are shown in Figs. 1, 2 and 3, respectively. In the three figures, a result of a modified greedy algorithm is labelled with the heuristic function the algorithm adopts, and a result of the “randomly choosing nodes” heuristic is labelled “Random”. For the CCP model, we set $\alpha$ and $\beta$ both as 0.9; while for the DCP model, the value of $\tau_0$ is set to be 12.

From Figs. 1, 2 and 3, we have some clear observations. No matter what value the budget is, the modified greedy algorithm generates better solutions than the “randomly choosing nodes” heuristic. The only expectation is that when solving the DCP model with $B = 50$, the modified greedy algorithm with heuristic function “SPL” has the same solution performance with the “randomly choosing nodes” heuristic. This is because there are at most two nodes in the target set when $B = 50$, and the power of the modified greedy algorithm may be weakened in searching such a small set. The second observation is that, no matter which algorithm is employed,
increasing the budget $B$ will improve the value of the objective for every model, which coincides with our intuition. In addition, we find that none of the three heuristic functions is better than the others for all the problems.

In the next experiment, we show how the modified greedy algorithm trades off between solution performance and computational complexity. We let $p = 0.02$ and $B = 100$ and consider the EVM model. We change the value of the trade-off parameter $r$ and obtain a number of results that are shown in Fig. 4. It is not difficult to find that the stochastic simulation is entirely avoided when $r = 1$. Another extreme is that when $r(t) = |N| - t$ depends on the volume of the current target set $t$, the modified greedy algorithm degenerates to the traditional greedy algorithm. We find that, in general, greater $r$ will make the modified greedy algorithm generate better solutions, however at the expense of spending more time. On the other hand, smaller $r$ helps save computation, however the performance of solutions is poorer. Therefore, Fig. 4 illustrates the fact that the modified greedy algorithm trades off between optimality and complexity.

The last experiment is performed to show the difference between the solutions of the problems with stochastic costs and those of the classic CIT minimization problem with determinate costs. We replace the cost for influencing each node in the first experiment by the expected value of the corresponding random variable, and consider the classic CIT minimization problem in which all the costs are determinate and the objective is the same with that in the EVM model. The results are labelled “Determinate costs”. In order to make the comparison sensible, we replace the objective in the CCP model with that of the EVM model. We consider different values of $\beta$ in the constraints in the CCP model and report the results that are labelled “Stochastic costs”. The results are shown in Fig. 5. We find that the solutions for the problem with stochastic costs are different from those for the classic problem with determinate costs. Furthermore, it is shown that increasing the value of $\beta$ makes the constraint more restricted and thus yields worse solutions.
Fig. 4. The solutions and runtime of the modified greedy algorithm.

Fig. 5. Comparison of the results for classic CIT minimization problems and those for the problems with stochastic costs.
6. Conclusions

In this paper, we have studied a particular problem on how to optimize the diffusion of influence in a social network. We have focused on the complete influence time minimization problem given that the available budget for initially influencing nodes is limited and that the cost for targeting each node is stochastic. The incremental chance model has been adopted to characterize the mechanism of the influence diffusion so as to guarantee the complete influence. Different decision criteria on optimality have been considered, and three different stochastic programming models have been proposed. In order to solve the proposed models, we have modified the traditional greedy algorithm so that the complexity and optimality can be better traded off. By performing experiments, we have found that the performance of the modified greedy algorithm we present dominates that of a trivial heuristic named “randomly choosing nodes”. Besides, we have empirically shown how the modified greedy algorithm helps decision makers get a good balance between solution performance and computational complexity.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 71101027) and Program for Young Excellent Talents, UIBE (No. 12YQ08).

References


