Information Processing Letters 114 (2014) 1–4

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On the inapproximability of minimizing cascading failures under the deterministic threshold model

Lei Nie [a,b,*], Jingjie Liu [a,b], Haicang Zhang [a,b], Zhiwei Xu [a]

[a] Institute of Computing Technology, Chinese Academy of Sciences, China
[b] University of Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 8 July 2012
Received in revised form 7 October 2013
Accepted 21 October 2013
Available online 25 October 2013
Communicated by J. Torán

Keywords:
Combinatorial problems
Inapproximability
Cascading failures
Deterministic threshold model

ABSTRACT

Given a directed graph G and a threshold $L(r)$ for each node r, the rule of deterministic threshold cascading is that a node r fails if and only if it has at least $L(r)$ failed in-neighbors. The cascading failure minimization problem is to find at most k edges to delete, such that the number of failed nodes is minimized. We prove an $n^{1-\epsilon}$ inapproximability result for the general case and a $\frac{1}{2}n^{\epsilon}$ inapproximability result for the special case with the maximum threshold of 1.

1. Introduction

Cascading failures are ubiquitous in various types of networks. For example, bankruptcies of several banks can shock other banks through a financial network and lead to a global financial crisis [1]. The spread of cascading failures is highly relevant to the topology of the propagation network [2]. There are several methods, such as blocking edges and vaccinating nodes, which can modify networks more resilient to cascading failures. This paper focuses on minimizing the number of cascading failures by deleting a bounded number of edges.

In this paper, we use the concept of *deterministic threshold* to model the cascading processes in a network [2,3]. Let $G = \langle V, E, L\rangle$ be a directed graph with thresholds, where V is a set of nodes, $E \subseteq V \times V$ is a set of directed edges and $L : V \to \mathbf{N}$ is a threshold function mapping each node to a nonnegative integer. We define the *maximum threshold* $L_{\max} = \max_{r\in V} L(r)$. The rule of deterministic threshold cascading is that a node r fails if and only if it has at least $L(r)$ failed in-neighbors. At time 0, a cascading process is triggered by nodes with thresholds of 0 (initial failed nodes). At time t, every node r checks whether it has at least $L(r)$ failed in-neighbors. If so, the node r will fail. The state of a node can only switch from alive to failed, but cannot switch from failed to alive. The process stops at time t if and only if there are no new failed nodes. We use $\Phi(V, E, L)$ to denote the set of failed nodes at the end of a cascading process. It can be proved that Φ is uniquely determined. We define the *cascading failure minimization* (*CFM*) problem as follows.

Problem 1 *(Cascading failure minimization).* Given a nonnegative integer k (budget) and a directed graph with thresholds $G = \langle V, E, L\rangle$, find at most k edges to delete, such that the number of failed nodes is minimized. That is,

$$\min_{E'\subseteq E} \left|\Phi\left(V, E - E', L\right)\right|$$
$$\text{s.t.}\quad |E'| \leqslant k.$$

We study the hardness of the cascading failure minimization problem and prove the first inapproximability

[*] Corresponding author.
E-mail addresses: nielei@ict.ac.cn (L. Nie), liujingjie@ict.ac.cn (J. Liu), zhanghaicang@ict.ac.cn (H. Zhang), zxu@ict.ac.cn (Z. Xu).

0020-0190/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ipl.2013.10.007

result by a polynomial time reduction from the vertex cover problem.

Theorem 1. *It is NP-hard to approximate the cascading failure minimization problem within a factor of $n^{1-\epsilon}$, for any $0 < \epsilon \leqslant 1$.*

The maximum threshold is equal to 2 in the above reduction, which motivates us to study the special case with the maximum threshold of 1. We prove the second inapproximability result by an approximation-preserving reduction from the *minimum coverage problem* [4].

Theorem 2. *Unless there exists a probabilistic algorithm for SAT that runs in time 2^{n^δ} on an instance of size n, there is no $\frac{1}{2}n^\epsilon$-approximation algorithm for the cascading failure minimization problem even if the maximum threshold equals 1, where $\delta > 0$ is an arbitrarily small constant and $\epsilon = 1/2^{O(1/\delta \log(1/\delta))}$.*

The cascading failure minimization problem with the maximum threshold of 1 is a special case of the *quarantining problem* [5] and the *minimum-size bounded-capacity cut problem* [6]. We can get a $\frac{1}{2}n^\epsilon$ inapproximability result of the above two problems as a corollary of Theorem 2. In addition, there are several related problems in which the initial failed nodes are not known in advance [7,8]. In contrast to minimize the spread of cascades, the influence maximization problem, motivated by viral marketing, aims to choose a set of k individuals to adopt a new product, such that the number of further adoptions through the cascade in social networks is maximized [9]. Several hardness results and approximation algorithms for this problem were reviewed by Kempe [10].

2. Proof of Theorem 1

We prove Theorem 1 by a polynomial time reduction from the NP-complete vertex cover problem. Given a vertex cover instance with a vertices, b edges and a nonnegative integer k, we can construct a CFM instance as follows. For each edge in the vertex cover instance, we add a node with a threshold of 2 in the CFM instance. We call them edge-nodes. For each vertex in the vertex cover instance, we add a node with a threshold of 1 in the CFM instance. We call them vertex-nodes. If a vertex and an edge are adjacent in the vertex cover instance, we produce a directed edge from the corresponding vertex-node to the corresponding edge-node in the CFM instance. Next, we add a node with a threshold of 0 in the CFM instance, which is called zero-node. We produce directed edges from the zero-node to each vertex-node. We add M nodes with thresholds of 1 in the CFM instance, which are called slave-nodes, where

$$M = \bigl(2(a+1)(1+2a+3b)\bigr)^{\frac{1}{\epsilon}}. \tag{1}$$

We produce directed edges from each edge-node to each slave-node. Finally, let the budget of the CFM instance be k given in the vertex cover instance. It is clear that this reduction can be completed in polynomial time. The length of the reduced CFM instance is $1 + 2a + 3b + (b + 1)M$, which is the total number of the nodes and edges. Fig. 1 shows an example of this reduction.

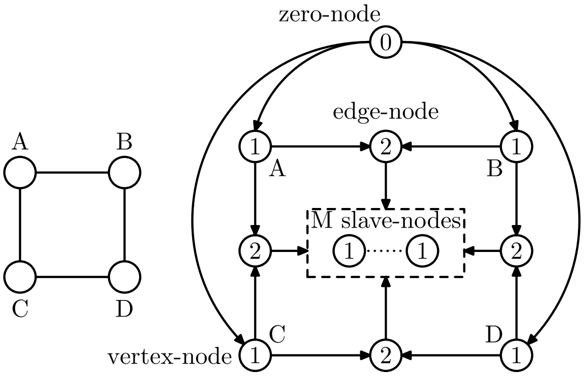

Fig. 1. An example of the reduction from a vertex cover instance (left) to a CFM instance (right).

Lemma 1. *Given a CFM instance reduced from a vertex cover instance, any optimal set of edges to delete only contains the edges from the zero-node to vertex-nodes.*

Proof. Because k is less than or equal to the number of vertices, the budget in the CFM instance is less than or equal to the number of edges from the zero-node to vertex-nodes. If there is an optimal set containing an edge from a vertex-node to an edge-node or an edge from an edge-node to a slave-node, we can replace it with an edge from the zero-node to a vertex-node, which will prevent at least one more node from failing. □

Lemma 2. *If the answer of a vertex cover instance is YES, then the OPT (optimal value) of the reduced CFM instance is $a-k+1$.*

Proof. Suppose that there is a k vertex cover. We delete the k edges from the zero-node to the vertex-nodes, corresponding vertices of which are in the vertex cover. Because every edge in the vertex cover instance has one or two endpoints in the vertex cover, all of the edge-nodes and the slave-nodes are alive. The number of failed nodes is $a - k + 1$. According to Lemma 1, any optimal set of edges to delete only contains the edges from the zero-node to vertex-nodes. It can save k vertex-nodes at most by deleting k edges from the zero-node to vertex-nodes. Thus, $a-k+1$ is the OPT. □

Lemma 3. *If the answer of a vertex cover instance is NO, then the OPT of the reduced CFM instance is greater than M.*

Proof. According to Lemma 1, any optimal set of edges to delete only contains the edges from the zero-node to vertex-nodes. Moreover, we cannot find a set of k edges from the zero-node to vertex-nodes in the reduced CFM instance to delete to make sure all edge-nodes are alive, because there is not any k vertex cover. The OPT is therefore greater than M, since all of the M slave-nodes will fail. □