Computer Science Review 28 (2018) 92–117

Contents lists available at ScienceDirect

Computer Science Review

journal homepage: www.elsevier.com/locate/cosrev

Survey

The Critical Node Detection Problem in networks: A survey

Mohammed Lalou[a,b,c,*], Mohammed Amin Tahraoui[c], Hamamache Kheddouci[c]

[a] Département d'Informatique, Faculté des Sciences Exactes, Université A. Mira Béjaia, 06000 Béjaia, Algeria
[b] Institut des Sciences et Technologie, Centre Universitaire A. Boussouf Mila, 43000 Mila, Algeria
[c] Université Claude Bernard Lyon 1-UFR-Informatique, Lab LIRIS, 43 bd du 11 Novembre 1918, F-69622 Villeurbanne Cedex, France

ARTICLE INFO

Article history:
Received 29 December 2016
Received in revised form 18 February 2018
Accepted 20 February 2018

Keywords:
Critical node
Important node
Node-deletion problem
Complex network
Combinatorial optimization
Complexity
Approximation schemes
Graph connectivity

ABSTRACT

In networks, not all nodes have the same importance, and some are more important than others. The issue of finding the most important nodes in networks has been addressed extensively, particularly for nodes whose importance is related to network connectivity. These nodes are usually known as *Critical Nodes*. The *Critical Node Detection Problem* (*CNDP*) is the optimization problem that consists in finding the set of nodes, the deletion of which maximally degrades network connectivity according to some predefined connectivity metrics. Recently, this problem has attracted much attention, and depending on the predefined metric, different variants have been developed. In this survey, we review, classify and discuss several recent advances and results obtained for each variant, including theoretical complexity, exact solving algorithms, approximation schemes and heuristic approaches. We also prove new complexity results and induce some solving algorithms through relationships established between different variants.

Contents

* Corresponding author at: Département d'Informatique, Faculté des Sciences Exactes, Université A.Mira Béjaia, Algeria.
E-mail addresses: mohammed.lalou@univ-lyon1.fr, m.lalou@centre-univ-mila.dz (M. Lalou), mohammed-amin.tahraoui@univ-lyon1.fr (M.A. Tahraoui), hamamache.kheddouci@univ-lyon1.fr (H. Kheddouci).

https://doi.org/10.1016/j.cosrev.2018.02.002
1574-0137/

1. Introduction

Many mechanisms and functions in networks are greatly affected by a fraction of nodes, which are usually qualified as *important*. A node is *important* if its failure or malicious behavior significantly degrades network performance. The issue of identifying the most important nodes in networks has long been the focus of an intensive amount of research. In the literature, these nodes appear under various names, depending on their role in the network, such as: most influential nodes [1], most vital nodes [2], most k-mediator nodes [3], key-player nodes [4]. With respect to network connectivity, the most important nodes are often known as *Critical Nodes*.

Therefore, critical nodes of a network are those the removal of which significantly degrades network connectivity. The *Critical Node Detection Problem* (*CNDP*) is the optimization problem that consists in finding the set of these nodes. In other words, the *CNDP* consists in finding the set of nodes whose deletion leads to achieving a certain objective related to making the network disconnected.

As a network can take different forms after disconnection, *node criticality* depends on how the network is disconnected once the node has been deleted, which depends, in turn, on the objective of the application considered. Thus, a node that is critical for some purposes or considerations may not be critical for others. Considering the network in Fig. 1, if we ask for the set of two nodes, the removal of which maximizes the number of connected components, then the optimal solution is to delete nodes $\{a, d\}$, which generates seven components. While if we consider the case where we aim at minimizing the set of nodes, the removal of which constrains the cardinality of each connected components to four nodes at most, the optimal solution is then to delete the nodes $\{b, c\}$. In both cases, we ask for the same thing: to delete nodes from the graph, but for different objectives. These objectives depend on the application at hand. For instance, the last case is relevant when inheriting the spreading of complex contagions[1] where more than four interactions are needed to acquire the contagion. In fact, if we assume that the graph in Fig. 1 corresponds to a social network, then vaccinating critical nodes $\{b, c\}$ (corresponding to individuals) allows the network to be partitioned on communities of four susceptible individuals each, at most (see Fig. 1). Hence either all individuals in the community are infected or not, there is no spread of the contagion as there is no chance of having more than four interactions with infected individuals.

Therefore, we can define critical nodes as those the deletion of which disconnects the network according to some predefined connectivity metrics, such as: maximizing the number of connected components, minimizing pairwise connectivity in the network, minimizing the largest component size, etc.

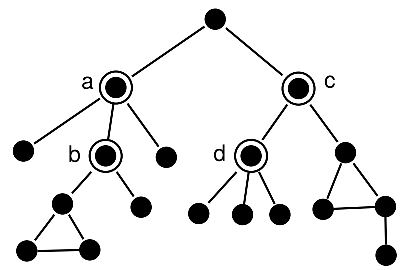

Fig. 1. Identifying critical nodes in a network with respect to the objective of the application considered.

Graph connectivity has long been considered as the measure of network robustness, and assessing how well the graph is connected or, on the contrary, stating how much effort (by deleting nodes or edges) is required to disconnect it has been extensively studied in the literature. This problem is known as the edge- or vertex-connectivity problem. For a detailed review, we refer the reader to [5]. Generally, deleting nodes (or edges) from a graph in order to obtain a disconnected graph with some specific properties is one of the oldest issues in graph theory. For problems based on the deletion of edges, a variety of variants have been defined and studied in the literature, such as the graph partitioning problem [6–8], the minimum k-cut problem [9,10], the multicut problem [11–13], the multiway cut problem [14–16], the multi-multiway cut problem [17], etc. For problems based on the deletion of nodes, which include the problem we are reviewing in this paper, we ask for deleting nodes rather than edges. Many variants have been studied in the literature, including:

- *The vertex separator problem.* Given a graph $G = (V, E)$ and an integer k, this problem aims at partitioning G into three subsets of nodes A, B and C such that $|C|$ is minimum, A and B are disconnected and $max(|A|, |B|) < k$. In other words, it seeks to find the minimum set of nodes $C \subseteq V$, the deletion of which partitions the graph into two bounded subsets A and B of at most k nodes. The problem is NP-hard even on 3-bounded degree graphs[2] [18], and it has been studied on general graphs [18], planar graphs [19,20], using polyhedral approach [21,22], metaheuristic algorithms [23], etc.
- *The multi-terminal vertex separator problem.* Given a graph $G = (V, E)$ and k terminal nodes, this problem consists in finding a subset $S \subseteq V$ of non-terminal nodes of minimum weight, the deletion of which generates k components, each

[1] This is a kind of contagion where an individual in the network acquires the contagion through interaction with $n > 1$ other affected individuals (n is an integer).

[2] A graph G is k-bounded degree if the degree of each node is less than or equal to k.