

Critical Node Detection Problem over a Path

Syed Md Omar Faruk

Department of Mathematics, Shahjalal University of Science and Technology, Sylhet, Bangladesh

Abstract In this paper we consider the problem of identifying a limited amount of nodes from a given graph in order to minimize some measure of the connectivity of the surviving graph. These nodes are commonly referred to as critical nodes. In particular, we look at the problem of identifying important nodes in a path graph whose removal optimizes the connectedness of the given path. We analyze four variants of the critical node detection problem on paths and present a closed-form solution for the objective function's optimality.

Keywords Critical node detection, Pairwise connectivity, Polynomial time algorithm

1. Introduction

Critical node detection problems (CNDP) are a family of optimization problems defined on graphs, where one is required to select a small set of nodes to remove in order to make the remaining graph as disconnected as possible, according to some connectivity measure. In particular, the connectivity measure analyzed in this paper is the so-called pairwise connectivity, formalized in [5], which counts the number of node pairs that are connected to each other by a path. This class of problems has attracted the interest of many researchers in the last two decades, because of its relevance in a number of practical applications. The applications involve scenarios in which the goal is to either secure the most critical nodes in a network or attack the most critical nodes in order to have the fewest connections between any two nodes in the network. Below are some of the most important applications of critical node detection problems.

The connections between pairs of nodes are minimized after the most critical nodes are removed from a supply chain network. A military supply chain network [33], for example, consists of battalions and support battalions as nodes and the connections between them as links. By attacking the most critical nodes in this network, the connectivity between supply and demand nodes will be reduced. As a result, solving the critical node detection problem (CNDP) is critical in military tactical attacks during wars.

Terrorists can be represented as nodes in a graph using information obtained from a covert network, with social interactions between them as links. We can reduce terrorist communication by targeting the most critical individuals in the networks [18].

In order to analyze the influence of epidemics in genuine social networks [22], people and their connections are represented as nodes and links in a graph. Since random mass vaccines are expensive [22], numerous solutions for targeted vaccinations were offered to reduce the spread of infectious illnesses in genuine social networks. However, the best vaccine technique is to identify and vaccinate important nodes in order to reduce pairwise connectivity between persons in a community [5], assuming that higher pairwise connectivity causes faster outbreaks.

Telecommunication networks, such as the Internet, telephone networks, and computer networks, can be visualized as graphs, with each node representing a terminal and links representing communication between terminals. In telecommunication networks, we aim to stop a virus from spreading or discover a strategy to decrease network [6] communication as much as feasible. Also, with network vaccination [19], where a graph reflecting individuals's contacts is provided, only a certain number of people can be vaccinated, and we intend to keep the virus [10,34] from spreading.

The CNDP's applications are not restricted to those listed above; it has a wide range of uses in a variety of fields, including transportation engineering [17], computational biology [24,32], the analysis of complex networks [8,14], security/vulnerability issues in networks [13,23,30,31], homeland security [9,15,16], energy [27], etc.

Since problems of this type usually turn out to be NP-hard for general graphs, and often also for quite special classes of graphs [2,11,28,29,30], several heuristic approaches as well as methods based on integer programming formulations have been proposed in the literature; see, e.g., [1,3,5,12,23,25,26]. Polynomial algorithms are instead available only for some particular cases, which are usually limited to graphs with bounded treewidth, in particular trees and series-parallel graphs [2,4,7,11,20,28]. We refer the interested reader to the survey [21] for a detailed overview.

* Corresponding author:

omarfaruk-mat@sust.edu (Syed Md Omar Faruk)

Received: Jun. 1, 2022; Accepted: Jun. 17, 2022; Published: Jun. 23, 2022

Published online at <http://journal.sapub.org/am>

2. The Problems that We Study

In this paper we study the Critical Node Detection Problem on a path which can be seen as a very special case of a tree. Given a path graph $P=(V,E)$ with $|V|=n$ nodes and an integer K , the CNDP on a path seeks to find a set $S \subseteq V$ of at most $K \leq n$ nodes, the deletion of which minimizes pairwise connectivity in the remaining path graph $P(V \setminus S)$. We will tackle the so-called Distance-based Critical Node Detection Problem (D-CNDP) over paths as described in [4]. The NP-completeness of D-CNDP over paths for some specific distance functions has been already established [4]. We look at some particular versions of CNDP on paths that fall into the D-CNDP on paths and provide closed-form solution for optimality of the objective function. The variants that we analyze in this paper are the following:

1. Minimizing the number of node-pairs connected by a path with upper bound of length at most L .
2. Minimizing the number of node-pairs connected by a path with lower bound of length at least L .
3. Minimizing the number of node-pairs connected by a path of length precisely L .
4. Minimizing the number of node-pairs connected by a path of length between L_1 and L_2 where L_1 and L_2 are the lower and upper bound respectively.

The term *balanced solution* is an important concept that we will utilize in the following statements. We call the solution balanced if the difference between the length of the shortest and longest components is either 0 or 1.

Before deriving the exact solution of the objective function for the four variants introduced above, we first present the following result without any lower or upper bound on the length of the connecting path.

Proposition 1. *For the CNDP on a path without any lower or upper bound, the optimal solutions are the balanced solutions.*

Proof. Let $P=(V,E)$ be a path graph with $|V|=n$ nodes and let α be the average number of nodes of the sub-paths obtain after removing K nodes, where $\alpha = \frac{|V|-K}{K+1}$.

Assume that we are given an optimal solution after

removing K nodes. Suppose by contradiction that in the given optimal solution, not all the sub-paths have length $\lfloor \alpha \rfloor$ or $\lceil \alpha \rceil$. Since the solution is unbalanced, there are at least two sub-paths in this solution in which one of the component has length at most $\lfloor \alpha \rfloor - 1$ and another component has length at least $\lceil \alpha \rceil + 1$. Let l_1 and l_2 be the number of nodes of the shortest and the longest components respectively, i.e., $l_1 \leq \lfloor \alpha \rfloor - 1$ and $l_2 \geq \lceil \alpha \rceil + 1$ (see Figure 1). Now we construct a new solution by adding one node in the shortest component, i.e., $l_1 + 1$ and by subtracting one node in the longest component, i.e., $l_2 - 1$ and the length of the other components are unchanged (see Figure 2). We want to show that this new solution is better than the given solution.

So the claim is

$$\begin{aligned} \binom{l_1+1}{2} + \binom{l_2-1}{2} &< \binom{l_1}{2} + \binom{l_2}{2}, \text{ where } l_2 - l_1 \geq 2 \\ \Leftrightarrow (l_1+1)l_1 + (l_2-1)(l_2-2) &< l_1(l_1-1) + l_2(l_2-1) \\ \Leftrightarrow l_1(l_1+1-l_1+1) &< (l_2-1)(l_2-l_2+2) \\ \Leftrightarrow 2l_1 &< 2(l_2-1) \\ \Leftrightarrow l_2 - l_1 &> 1 \end{aligned}$$

which is true by assumption because we are assuming $l_2 - l_1 \geq 2$, and we will continue this perturbation until we have $K+1$ components of size $\lfloor \alpha \rfloor$ and $\lceil \alpha \rceil$, and this shows that every optimal solution is balanced. Observe that all balanced solutions have the same cost and hence they are equivalent. Since all balanced solutions are equivalent, this proves that all balanced solutions are optimal.

We now state the result when the graph is a path and the problem is to remove K nodes in order to minimize the number of pairs that are connected by a path of length at most L , for some given natural number L .

Proposition 2. *For the CNDP on a path with upper bound L , any solution that removes K nodes and creates $K+1$ components with cardinality at least L is optimal. If no such a solution exists, then any solution that removes K nodes and creates balanced components is optimal.*

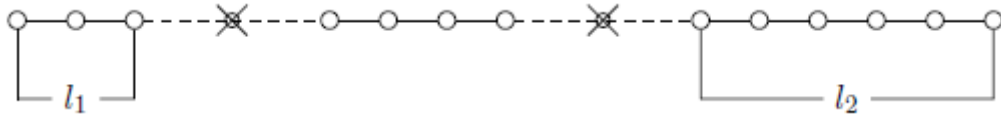


Figure 1. Example of a solution after removing $K = 2$ nodes

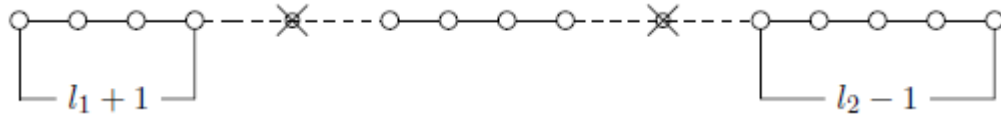


Figure 2. Modification of the solution of Figure 1

Proof. First we prove the second part of the statement which means there is no way to make all the components with cardinality at least L and in this case we want to prove that balanced solutions are optimal.

Assume by contradiction that we are given an optimal solution which is not balanced. Since we are assuming that there is no way to make all the components with cardinality at least L , there is at least one sub-path with cardinality at most $L-1$. Let A be the shortest component and B be the longest component of the solution. With a slight abuse of notation we will denote by A and B also the number of nodes in the component. Then $A \leq L-1$ and $B-A \geq 2$ because the solution is not balanced. Now we do the perturbation by adding one node in the shortest component and by subtracting one node in the longest component and by keeping the cardinality of all other components unchanged. Now if $B > L$ and we subtract a node from B , then we lose L connections, while if $B \leq L$ and we subtract a node from B , then we lose $(B-1)$ connections because we have to count only the paths of length $\leq L$. Since $A \leq L-1$, if we add an extra node to A , then we have A new connections. Thus we have the following two cases when we do the perturbation.

Case 1: If $B > L$, then the improvement is < 0 .

Case 2: If $B \leq L$, then the improvement is $A-B+1$ which becomes ≤ -1 because we are assuming that $B-A \geq 2$.

So with the property $A \leq L-1$, we have always a negative improvement (case 1 and 2), i.e., we reduce the objective function and this proves that any unbalanced solution is not optimal. Since all balanced solutions are equivalent, this proves that all balanced solutions are optimal.

Now we prove the first part of the statement which means that it is possible to make all the sub-paths with cardinality at least L . In this case still it is possible that in the solution the shortest component is smaller than $L-1$ and we know by the above argument that this solution is not optimal.

In the following we prove that all the solutions that creates $K+1$ sub-paths with cardinality at least L are equivalent.

When we remove K nodes from the path, we create $K+1$ components. Suppose t_i is the number of edges of every sub-path. Now if we take a path with $t_i \geq L-1$ edges,

then the contribution is given by $L(t_i - L + 1) + \sum_{K=1}^{L-1} K$,

because in a component of $(t+1)$ nodes, we have $(t+1-L)$ nodes which gives a contribution of length L and the remaining L nodes gives the contribution of $(L-1), (L-2), \dots, 1$, respectively, which is given by $\sum_{K=1}^{L-1} K$.

In the last component, the number of edges is given by

$t_{K+1} = n-1 - \sum_{i=1}^K t_i - 2K$, because we have $n-1$ edges initially, then we have to discard all the edges of the

sub-paths which is given by $\sum_{i=1}^K t_i$ and when we are

removing K nodes, we are deleting $2K$ edges, since we are assuming that we are creating $K+1$ components that means we are not removing two adjacent nodes. So the contribution of the $(K+1)$ th component is given by

$$L(n-1 - \sum_{i=1}^K t_i - 2K) + \sum_{K=1}^{L-1} K. \text{ Hence the total contribution is}$$

$$\sum_{i=1}^K L(t_i - L + 1) + \sum_{K=1}^{L-1} K + L(n-1 - \sum_{i=1}^K t_i - 2K)$$

$$+ \sum_{K=1}^{L-1} K = 2 \sum_{K=1}^{L-1} K + L(n-L-2K)$$

which is independent of t_i 's, that means it does not matter what is the length of the sub-paths as long as their cardinality is at least L .

Now we show that all the solutions that create all components with cardinality at least L are optimal. Define S^* as the set of solutions which have the property that all components have at least L nodes. Since there is at least one solution with all the components with cardinality at least L , the set S^* is nonempty. Take a solution in S^* . This solution has the property that the total number of nodes in the surviving components is at least $(L-1)(K+1)$ as after removing K nodes we have $(K+1)$ components and each of cardinality at least $(L-1)$. Let's call again A and B be the shortest component and the longest component respectively in S^* . Take any solution which is not in S^* , which means that it has at least one component whose cardinality must be smaller than $L-1$, so that $A \leq L-2$. We claim that the cardinality of the longest component is at least L , i.e., $B \geq L$. If this is not true, then $B \leq L-1$. In this case the total number of nodes in the surviving components is at most $(L-2) + (L-1)K$, which is at least one unit smaller than $(L-1)(K+1)$ and this is a contradiction because we know that the total number of nodes can not be changed as we always remove K nodes. So any solution which is not in S^* has the property that $A \leq L-2$ and $B \geq L$ and this means that the solution is not balanced. Since the solution is not balanced, we apply the perturbation and by the previous arguments (case 1 and case 2) we know that when we do the perturbation we always improve. After the perturbation either we find a solution which is in S^* and we are done, or we find a solution which has still the property that $A \leq L-2$ and $B \geq L$ and we will continue this perturbation until the solution is balanced. When we get a balanced solution, we are in S^* and this proves that the set S^* is optimal.

The following result holds for the case when the graph is a path and the problem calls for minimizing the number of node-pairs still connected by a path of length at least L , for

some given natural number L , surviving in a path after having removed at most K nodes..

Proposition 3. *For the CNDP on a path with lower bound L , any solution that removes K nodes and creates $K+1$ components with cardinality at most L is optimal. If no such a solution exists, then any solution that removes K nodes and creates balanced components is optimal.*

Proof. First consider that we can make all the components with cardinality at most L and in this case all the solutions are equivalent and hence optimal, because the objective value is 0 and in this incident, it does not matter whether the solution is balanced or not as long as all the components have cardinality at most L .

Now we prove the second part of the statement which means there is no way to make all the components with cardinality at most L and in this case we want to prove that balanced solutions are optimal.

Assume by contradiction that we are given an optimal solution which is not balanced. Since we are assuming that there is no way to make all the components with cardinality at most L , there is at least one sub-path with cardinality greater than L . Let A be the shortest component and B be the longest component of the solution. With a slight abuse of notation we will denote by A and B also the number of nodes in the component. Then $B > L$ and $A \leq B-2$ because the solution is not balanced. Now we do the perturbation by adding one node in the shortest component and by subtracting one node in the longest component and by keeping the cardinality of all other components unchanged. Now if $A \geq L-1$ and we add an extra node to A , then we have $A-L+1$ new connections while if $A < L-1$ and we add an extra node to A , in this case we create 0 connection, because we have to count only the paths of length $\geq L$. Since $B > L$ and we subtract a node from B , we lose $B-L$ connections. Thus we have two cases when we do the perturbation.

Case 1: If $A \geq L-1$, then the improvement is $A-B+1$ which becomes ≤ -1 because we are assuming that $A \leq B-2$.

Case 2: If $A < L-1$, then the improvement is $L-B$ which becomes ≤ -1 because $B > L$.

So when we have at least one sub-path with cardinality at least L , we always get a better solution and this proves that any unbalanced solution is not optimal. Since all balanced solutions are equivalent, this proves that all balanced solutions are optimal.

When the graph is a path, and we want to minimize the number of node-pairs still connected by a path of precise length L , for some given natural number L , surviving in a path after having removed at most K nodes, the following result holds.

Proposition 4. *For the CNDP on a path of precise length L , any solution that removes K nodes and creates $K+1$ components with cardinality at least L is optimal. Otherwise, any solution that creates $K+1$ components with cardinality at most L is optimal. If no such a solution exists, then any solution that removes K nodes and creates balanced components is optimal.*

Proof. Assume by contradiction that we are given an optimal solution which is not balanced. We are assuming that there is no way to make all the components with cardinality at least L or all the components with cardinality at most L . Then there is at least one sub-path with cardinality less than L and at least one sub-path with cardinality greater than L . Let A be the shortest component and B be the longest component of the solution. With a slight abuse of notation we will denote by A and B also the number of nodes in the component. Then $A < L$, $B > L$ and $A \leq B-2$ because the solution is not balanced. Now we do the perturbation by adding one node in the shortest component and by subtracting one node in the longest component and by keeping the cardinality of all other components unchanged. Since $A < L$, when we add an extra node to A , we create 0 connection, because we have to count only the paths of length precisely L . Again since $B > L$, when we subtract a node from B , we lose 1 connection. Thus we have the following case when we do the perturbation.

Case 1: If $A < L$ and $B > L$, then the improvement is -1 .

In the following we prove that all the solutions that creates $K+1$ sub-paths with cardinality at least L are equivalent.

When we remove K nodes from the path, we create $K+1$ components. Suppose t_i is the number of edges of every sub-path. Now if we take a path with $t_i \geq L-1$ edges, then the contribution is given by $(t_i - L + 1)$, because in a component of $(t+1)$ nodes, we have $(t+1-L)$ nodes which gives a contribution of length exactly L . In the last component, the number of edges is given by

$$t_{K+1} = n - 1 - \sum_{i=1}^K t_i - 2K, \text{ because we have } n-1 \text{ edges}$$

initially, then we have to discard all the edges of the sub-paths which is given by $\sum_{i=1}^K t_i$ and when we are

removing K nodes, we are deleting $2K$ edges, since we are assuming that we are creating $K+1$ components that means we are not removing two adjacent nodes. So the contribution of the $(K+1)$ th component is given by

$$(n - 1 - \sum_{i=1}^K t_i - 2K). \text{ Hence the total contribution is}$$

$$\sum_{i=1}^K (t_i - L + 1) + (n - 1 - \sum_{i=1}^K t_i - 2K) = n - L - 2K$$

which is independent of t_i 's, that means it does not matter what is the length of the sub-paths as long as their cardinality is at least L and this proves that all the solutions that creates long components are all equivalent.

Now we show that all the solutions that create all components with cardinality at least L are optimal. Define S^* as the set of solutions which have the property that all components have at least L nodes. Since there is at least one solution with all the components with cardinality at least

L , the set S^* is nonempty. Take a solution in S^* . This solution has the property that the total number of nodes in the surviving components is at least $(L-1)(K+1)$ as after removing K nodes we have $(K+1)$ components and each of cardinality at least $(L-1)$. Let's call again A and B be the shortest component and the longest component respectively in S^* . Take any solution which is not in S^* , which means that it has at least one component whose cardinality must be smaller than $L-1$, so that $A \leq L-2$. We claim that the cardinality of the longest component is at least L , i.e., $B \geq L$. If this is not true, then $B \leq L-1$. In this case the total number of nodes in the surviving components is at most $(L-2)+(L-1)K$, which is at least one unit smaller than $(L-1)(K+1)$ and this is a contradiction because we know that the total number of nodes can not be changed as we always remove K nodes. So any solution which is not in S^* has the property that $A \leq L-2$ and $B \geq L$ and this means that the solution is not balanced. Since the solution is not balanced, we apply the perturbation and by the previous arguments (case 1) we know that when we do the perturbation we always improve. After the perturbation either we find a solution which is in S^* and we are done, or we find a solution which has still the property that $A \leq L-2$ and $B \geq L$ and we will continue this perturbation until the solution is balanced. When we get a balanced solution, we are in S^* and this proves that the set S^* is optimal.

Similarly, if we can make component in such a way that the cardinality of all components is shorter than L , then all the solutions are equivalent and hence optimal, because the objective value is 0 and in this incident, it does not matter whether the solution is balanced or not as long as all the components have cardinality at most L .

Otherwise, if we can not make solution in which all the components have cardinality at least L or at most L , then obviously we are in case 1, because case 2 is not possible and in this case we always have a negative improvement, i.e., we reduce the objective function and this proves that any unbalanced solution is not optimal. Since all balanced solutions are equivalent, this proves that all balanced solutions are optimal.

In the following we state the result when the graph is a path and the problem is to remove K nodes in order to minimize the number of pairs that are connected by a path of length at least L_1 and at most L_2 .

Now we consider the case in which we have a lower bound L_1 and an upper bound L_2 and we consider the connections which are between L_1 and L_2 .

Proposition 5. *For the CNDP on a path with lower bound L_1 and upper bound L_2 , any solution that removes K nodes and creates $K+1$ components with cardinality at least L_2 is optimal. Otherwise, any solution that creates $K+1$ components with cardinality at most L_1 is optimal.*

If no such a solution exists, then any solution that removes K nodes and creates balanced components is optimal.

Proof. Assume by contradiction that we are given an optimal solution which is not balanced. We are assuming that there is no way to make all the components with cardinality at least L_2 or all the components with cardinality at most L_1 . Let A be the shortest component and B be the longest component of the solution. With a slight abuse of notation we will denote by A and B also the number of nodes in the component. Then $A < L_2$, $B > L_1$ and $A \leq B-2$ because the solution is not balanced. Now we do the perturbation by adding one node in the shortest component and by subtracting one node in the longest component and by keeping the cardinality of all other components unchanged. Since we want to count the paths of length between L_1 and L_2 , if $A < L_1-1$ and we add an extra node to A , then we have 0 new connection while if $L_1-1 \leq A < L_2$ and we add an extra node to A , in this case we create $A-L_1+1$ connections. Again if $L_1 < B < L_2$ and we subtract a node from B , then we lose $B-L_1$ connections while if $B > L_2$ and we subtract a node from B , then we lose L_2-L_1 connections. Thus we have the following cases when we do the perturbation.

Case 1: If $A < L_1-1$ and $L_1 < B < L_2$, then the improvement is L_1-B which becomes ≤ -1 because $L_1 < B$.

Case 2: If $A < L_1-1$ and $B > L_2$, then the improvement is L_1-L_2 which becomes ≤ -1 since $L_1 < L_2$.

Case 3: If $L_1-1 \leq A < L_2$ and $L_1 < B < L_2$, then the improvement is $A-B+1$ which becomes ≤ -1 because we are assuming that $A \leq B-2$.

Case 4: If $L_1-1 \leq A < L_2$ and $B > L_2$, then the improvement is $A-L_2+1$ which becomes ≤ -1 because $A < L_2$.

If we can make all the components with cardinality at most L_1 , then all the solutions are equivalent and hence optimal, because the objective value is 0.

In the following first we prove that all the solutions that creates $K+1$ sub-paths with cardinality at least L_2 are equivalent.

When we remove K nodes from the path, we create $K+1$ components. Suppose t_i is the number of edges of every sub-path. Now if we take a path with $t_i \geq L_2-1$ edges, then the contribution is given by

$$(L_2-L_1+1)(t_i-L_2+1) + \sum_{K=1}^{L_2-L_1} K,$$

because in a component of (t_i+1) nodes, we have (t_i+1-L_2) nodes which gives a contribution of (L_2-L_1+1) and the remaining L_2-1 nodes gives the contribution of $(L_2-L_1), (L_2-L_1-1), \dots, 1$,

respectively, which is given by $\sum_{K=1}^{L_2-L_1} K$. In the last component, the number of edges is given by $t_{K+1} = n-1 - \sum_{i=1}^K t_i - 2K$, because we have $n-1$ edges initially, then we have to discard all the edges of the sub-paths which is given by $\sum_{i=1}^K t_i$ and when we are removing K nodes, we are deleting $2K$ edges, since we are assuming that we are creating $K+1$ components that means we are not removing two adjacent nodes. So the contribution of the $(K+1)$ th component is given by $(L_2 - L_1 + 1)(n-1 - \sum_{i=1}^K t_i - 2K) + \sum_{K=1}^{L_2-L_1} K$. Hence the total contribution is

$$\begin{aligned} & \sum_{i=1}^K (L_2 - L_1 + 1)(t_i - L_2 + 1) + \sum_{K=1}^{L_2-L_1} K \\ & + (L_2 - L_1 + 1)(n-1 - \sum_{i=1}^K t_i - 2K) + \sum_{K=1}^{L_2-L_1} K \\ & = 2 \sum_{K=1}^{L_2-L_1} K + (L_2 - L_1 + 1)(n - L_2 - 2K) \end{aligned}$$

which is independent of t_i 's, that means it does not matter what is the length of the sub-paths as long as their cardinality is at least L_2 and this proves that all the solutions that creates long components are all equivalent.

Now we show that all the solutions that create all components with cardinality at least L_2 are optimal. Define S^* is the set of solutions which has the property that all components have at least L_2 nodes. Since there is at least one solution with all the components with cardinality at least L_2 , the set S^* is nonempty. Take a solution in S^* . This solution has the property that the total number of nodes in the surviving components is at least $(L_2 - 1)(K+1)$ as after removing K nodes we have $(K+1)$ components and each of cardinality at least $(L_2 - 1)$. Let's call again A and B be the shortest component and the longest component respectively in S^* . Take any solution which is not in S^* , which means that it has at least one component whose cardinality must be smaller than $L_2 - 1$, so that $A \leq L_2 - 2$. We claim that the cardinality of the longest component is at least L_2 , i.e., $B \geq L_2$. If this is not true, then $B \leq L_2 - 1$. In this case the total number of nodes in the surviving components is at most $(L_2 - 2) + (L_2 - 1)K$, which is at least one unit smaller than $(L_2 - 1)(K+1)$ and this is a contradiction because we know that the total number of nodes can not be changed as we always remove K nodes.

So any solution which is not in S^* has the property that $A \leq L_2 - 2$ and $B \geq L_2$ and this means that the solution is not balanced. Since the solution is not balanced, we apply the perturbation and by the previous arguments (case 2 and case 4) we know that when we do the perturbation we always improve. After the perturbation either we find a solution which is in S^* and we are done, or we find a solution which has still the property that $A \leq L_2 - 2$ and $B \geq L_2$ and we will continue this perturbation until the solution is balanced. When we get a balanced solution, we are in S^* and this proves that the set S^* is optimal.

Otherwise, if we can not make solution in which all the components have cardinality at least L_2 or at most L_1 , then we are in one of the four cases and in every case we always have a negative improvement, i.e., we reduce the objective function and this proves that any unbalanced solution is not optimal. Since all balanced solutions are equivalent, this proves that all balanced solutions are optimal.

3. Conclusions

In this paper we focused our study on identifying critical nodes from a graph, whose deletion results in the minimum pairwise connectivity among the remaining nodes. We studied a particular case of the critical node detection problem, namely the Distance-based Critical Node Detection Problem (D-CNDP) over paths. We considered four versions of the D-CNDP on paths and provided closed-form solution to solve the problems.

REFERENCES

- [1] Addis, B., Aringhieri, R., Grosso, A., Hosteins, P.: *Hybrid constructive heuristics for the critical node problem*, *Annals of Operations Research*, 238(1-2), 637–649 (2016).
- [2] Addis, B., Di Summa, M., Grosso, A.: *Identifying critical nodes in undirected graphs: Complexity results and polynomial algorithms for the case of bounded treewidth*, *Discrete Applied Mathematics*, 161(16), 2349–2360 (2013).
- [3] Aringhieri, R., Grosso, A., Hosteins, P., Scatamacchia, R.: *Local search metaheuristics for the critical node problem*, *Networks*, 67(3), 209–221 (2016).
- [4] Aringhieri, R., Grosso, A., Hosteins, P., Scatamacchia, R.: *Polynomial and pseudopolynomial time algorithms for different classes of the distance critical node problem*, *Discrete Applied Mathematics*, 253, 103–121 (2019).
- [5] Arulselvan A., Commander C.W., Eleftheriadou L., Pardalos P.M.: *Detecting critical nodes in sparse graphs*, *Computers & Operations Research*, 36, 2193–2200 (2009).
- [6] Arulselvan, A., Commander, C. W., Pardalos, P. M. and Shylo, O.: *Managing network risk via critical node identification*, Gulpinar, N., Rustem, B. (eds.) *Risk Management in Telecommunication Networks*, 2010.

- [7] Berger, A., Grigoriev, A., van der Zwaan, R.: *Complexity and approximability of the k-way vertex cut*, Networks, 63(2), 170–178 (2014).
- [8] Borgatti, S.P.: *Identifying sets of key players in a network*, Computational and Mathematical Organization Theory, 12, 21–34 (2006).
- [9] Brown, G., Carlyle, M., Salmerón, J., Wood, K.: *Defending critical infrastructure*, Interfaces, 36(6), 530–544 (2006).
- [10] Cohen, R., Ben Avraham, D., Havlin, S.: *Efficient immunization strategies for computer networks and populations*, Physical Review Letters, 91, 247901–247905 (2003).
- [11] Di Summa, M., Grosso, A., Locatelli, M.: *Complexity of the critical node problem over trees*, Computers & Operations Research, 38(12), 1766–1774 (2011).
- [12] Di Summa, M., Grosso, A., Locatelli, M.: *Branch and cut algorithms for detecting critical nodes in undirected graphs*, Computational Optimization and Applications, 53(3), 649–680 (2012).
- [13] Dinh, T., Xuan, Y., Thai M.T. Park, E., Znati, T.: *On approximation of new optimization methods for assessing network vulnerability*, In: Proceedings of INFOCOM, IEEE (2010).
- [14] Fan, N., Pardalos, P.: *Robust optimization of graph partitioning and critical node detection in analyzing networks*, In: W.Wu, O. Daescu (eds.) International Conference on Combinatorial Optimization and Applications, Lecture Notes in Computer Science, vol. 6508, pp. 170–183. Springer (2010).
- [15] Grubestic, T.H., Matisziw T.C., Murray, A.T., Snediker D.: *Comparative approaches for assessing network vulnerability*, Int. Reg. Sci. Rev., 31(1), 88–112 (2008).
- [16] Houck, D.J., Kim, E., O'Reilly, G.P., Picklesimer, D.D., Uzunalioglu, H.: *A network survivability model for critical national infrastructures*, Bell Labs Technical J., 8(4), 153–172 (2004).
- [17] Jenelius, E., Petersen, T., Mattsson, L.G.: *Importance and exposure in road network vulnerability analysis*, Transportation Research Part A: Policy and Practice 40(7), 537–560 (2006).
- [18] Krebs V.: *Uncloaking terrorist networks*, First Monday, 7 (2002).
- [19] Kuhlman, C., Kumar, V., Marathe, M., Ravi, S., Rosenkrantz, D.: *Finding critical nodes for inhibiting diffusion of complex contagions in social networks*. In: J. Balczar, F. Bonchi, A. Gionis, M. Sebag (eds.) Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science, vol. 6322, pp. 111–127. Springer (2010).
- [20] Lalou, M., Tahraoui, M., Kheddouci, H.: *Component-cardinality-constrained critical node problem in graphs*, Discrete Applied Mathematics, 210, 150–163 (2016).
- [21] Lalou, M., Tahraoui, M., Kheddouci, H.: *The critical node detection problem in networks: A survey*, Computer Science Review, 28, 92–117 (2018).
- [22] Miller, J. C., and Hyman, J. M.: *Effective vaccination strategies for realistic social networks*, Physica A: Statistical Mechanics and its Applications, 386(2): 780–785, (2007).
- [23] Nguyen, D., Shen, Y., Thai, M.: *Detecting critical nodes in interdependent power networks for vulnerability assessment*, IEEE Transactions on Smart Grid, 4(1), 151–159 (2013).
- [24] Pawson, T., Nash, P.: *Protein–protein interactions define specificity in signal transduction*, Genes Dev. 14(9), 1027–1047 (2000).
- [25] Pullan, W.: *Heuristic identification of critical nodes in sparse real-world graphs*, Journal of Heuristics, 21(5), 577–598 (2015).
- [26] Purevsuren, D., Cui, G., Win, N., Wang, X.: *Heuristic algorithm for identifying critical nodes in graphs*, Advances in Computer Science: an International Journal, 5(3), 1–4 (2016).
- [27] Salmeron, J., Wood, K.R., Baldick, R.: *Analysis of electric grid security under terrorist threat*, IEEE Transactions on Power Systems, 19(2), 905–912 (2004).
- [28] Shen, S., Smith, J.: *Polynomial-time algorithms for solving a class of critical node problems on trees and series-parallel graphs*, Networks, 60(2), 103–119 (2012).
- [29] Shen, S., Smith, J., Goli, R.: *Exact interdiction models and algorithms for disconnecting networks via node deletions*, Discrete Optimization, 9(3), 172–188 (2012).
- [30] Shen, Y., Nguyen N.P. Xuan, Y., Thai, M.: *On the discovery of critical links and nodes for assessing network vulnerability*, IEEE/ACM Transactions on Networking, 21(3), 963–973 (2013).
- [31] Veremyev, A., Prokopyev, O., Pasiliao, E.: *Critical nodes for distance-based connectivity and related problems in graphs*, Networks, 66(3), 170–195 (2015).
- [32] Yan, C., Wu, F., Jernigan, R.L., Dobbs, D., Honavar, V.: *Characterization of protein–protein interfaces*, Protein J. 27 (1), 59–70 (2008).
- [33] Zhao, K., Kumar, A., Harrison, T. P. and Yen, J.: *Analyzing the resilience of complex supply network topologies against random and targeted disruptions*, IEEE Systems Journal, 5(1), 28–39 (2011).
- [34] Zhou T., Fu Z.-Q., Wang B.-H.: *Epidemic dynamics on complex networks*, Progress in Natural Science, 16, 452–457 (2006).