MŰEGYETEM 1782

Budapest University of Technology and Economics

Institute of Mathematics

Department of Stochastics

MSc Thesis

Extremal problems of color-avoiding connectivity

József Pintér

Supervisors:

Dr. Roland Molontay

Department of Stochastics

Budapest University of Technology and Economics

Dr. Kitti Varga

Department of Computer Science and Information Theory

Budapest University of Technology and Economics

2022

Contents

1 Introduction

Network theory is an exceptionally important interdisciplinary field, whose importance has also been demonstrated in recent years, since it played a prominent role in curbing the pandemic [16]. Network theory is also used in a much more wider range of areas than just epidemic control, for example communication technology, public transport, supply chains, etc., since complex networks can efficiently describe systems examined in different areas [2].

An important aspect is the resistance of the networks against failure, i.e., whether a network can withstand a specific attack or not. A usual type of attacks is the removal of some edges or vertices of the given network. These attacks can happen randomly or they might be targeted on a special subset of edges or vertices.

Although the standard frameworks of error/attack tolerance in complex networks can be really useful in industrial practices, we might be able to develop a more efficient framework if we take into account that some parts of the network share some vulnerabilities. A characteristic example is the case of public transport networks, where the edges have their corresponding vehicle types such as buses, trams, trains or subways. One can observe that excessive snowing or snowstorms have a greater impact on the railway transport than on subways. In extreme cases these weather conditions might even paralyze the railway traffic, which we can think of (from a network theoretical point of view) that all the edges corresponding to railway vehicles disappear from the network. If we color the edges of the complex network according to their vehicle types, then this scenario would lead us to the theory of color-avoiding connectivity.

The concept of color-avoiding connectivity, which studies the resistance of colored networks, was introduced by Krause et al [11]. For example, given an edge-colored graph, we can check whether the graph remains connected if we remove the edges of a specific color. We call a graph edge-color-avoiding connected if with the removal of edges of any single color the graph remains connected, i.e., edge-color-avoiding connected graphs are completely resistant against color-specific attacks. We can also color the vertices of a graph instead of its edges, and consider a similar concept. In this case we can decide whether we want to remove those vertices as well, for which we are checking if they are color-avoiding connected, when we are removing vertices of their colors. This yields the definition of vertex-color-avoiding connectivity and internally vertex-color-avoiding connectivity.

In this thesis we explore the concept of color-avoiding connectivity and some related extremal problems. The contribution of the thesis can be summarized as follows:

- We prove that the minimum number of edges in an edge-color-avoiding connected graph, where the edges are colored with exactly $k \geq 2$ colors, is $n - 1 + \left\lceil \frac{n-1}{k-1} \right\rceil$ and if $k = 1$, then there exists no edge-color-avoiding connected graph on $n \geq 2$ vertices (see Theorem 4.1.1).

- We prove that the minimum number of edges in a vertex-color-avoiding connected graph, where the vertices are colored with exactly $k \geq 3$ colors, is n and if $k = 1$ or $k = 2$, then the minimum number of edges is $n - 1$ (see Theorem 4.2.1).

- We prove that the minimum number of edges in an internally vertex-color-avoiding connected graph, where the vertices are colored with exactly $k \geq 2$ colors, is $\left\lceil \frac{2k-1}{2k-2} n - \frac{k}{k-1} \right\rceil$ and if $k = 1$, then it is $\binom{n}{k}$ (see Theorem 4.3.1).

- We prove that finding an internally vertex-, vertex- or edge-color-avoiding connected spanning subgraph with minimum number of edges is NP-hard (see Theorem 5.1.1).

- We develop polynomial time approximation algorithms for finding an internally vertex-, vertex- or edge-color-avoiding connected spanning subgraph with minimum number of edges with approximation factors of $3\frac{2k-2}{2k-1}$, 2 and $\frac{2(k-1)}{k}$, respectively (see Theorems 5.2.3, 5.2.2 and 5.2.1).

In Section 2 we explain the definitions of color-avoiding connectivity and investigate some simple properties, and also provide a literature review. In Section 3 we present some useful definitions, notation and also some extremal problems which are closely related to the color-avoiding problems we consider. In Section 4 we prove the theorems about the minimum number of edges in an edge-, vertex- and internally vertex-color-avoiding connected graph. In Section 5 we prove the NP-hardness of finding an internally vertex-, vertex- or edge-color-avoiding connected spanning subgraph with minimum number of edges. In this section we also present the polynomial time approximation algorithms and prove the corresponding theorems about their running time and approximations factors.

2 Color-avoiding connectivity on complex networks

2.1 Definitions and notation

The color-avoiding connectivity framework was developed by Krause et al. [9, 11, 12]. The main idea of this theory is to treat together the edges or vertices of a complex network with common, shared vulnerabilities. A usual way to represent groups of edges or vertices of a complex network is to color them according to their groups. In the simplest version we assign exactly one color to every edge or vertex.

Definition 2.1. Let $G = (V, E)$ be a graph, $C = \{c_1, ..., c_k\}$ a set of colors and $c : E \to C$ a function that assigns colors to the edges.

The vertices $u, v \in V$ are called **edge-c_i-avoiding connected** if there exist a path between u and v which does not contain any edge of color c_i. Such paths are called **c_i-avoiding paths**. If two vertices are edge-c_i-avoiding connected for all $c_i \in C$, then we say that they are **edge-color-avoiding connected**.

All of these definitions related to the edge-color-avoiding connectivity can be defined similarly for vertices as well. But there is one very important aspect which leads to different definitions, thus to different problems. We can decide whether we want to remove those vertices as well, for which we are checking if they are color-avoiding connected, when we are removing vertices of their colors. This yields the definition of vertex-color-avoiding connectivity and internally vertex-color-avoiding connectivity.

Definition 2.2. Let $G = (V, E)$ be a graph, $C = \{c_1, ..., c_k\}$ a set of colors and $c : V \to C$ a function that assigns colors to the vertices.

The vertices $u, v \in V$ are called **vertex-c_i-avoiding connected** if either there exists a path between u and v, which does not contain any vertices of color c_i (these paths are called **vertex-c_i-avoiding paths**), or at least one of u or v is of color c_i. If two vertices are vertex-c_i-avoiding connected for all $c_i \in C$, then we say that they are **vertex-color-avoiding connected**.

Definition 2.3. Let $G = (V, E)$ be a graph, $C = \{c_1, ..., c_k\}$ a set of colors and $c : V \to C$ a function that assigns colors to the vertices.

The vertices $u, v \in V$ are called **internally vertex-c_i-avoiding connected** if there exists a path between u and v, which does not contain any internal vertices of color c_i (these paths are called **internally vertex-c_i-avoiding paths**). If two vertices are internally vertex-c_i-avoiding connected for all $c_i \in C$, then we say that they are **internally vertex-color-avoiding connected**.

Definition 2.4. Let $G = (V, E)$ be a graph. We say that G is internally vertex-, vertex- or edge-color-avoiding connected if it is connected, and any two vertices of G are internally vertex-, vertex- or edge-color-avoiding connected, respectively.

In the following definitions, we follow the terminology of Molontay and Varga [14] with some slight changes. For example, instead of of color-avoiding edge-connectivity, here we write edge-color-avoiding connectivity, similarly in the vertex-color-avoiding case. Also, weakly and strongly vertex-color-avoiding connectivity is replaced by vertex- and internally vertex-color-avoiding connectivity in this thesis.

In Figure 1 we can see examples for edge-color-avoiding connectivity and in Figure 2 we can see examples for vertex- and internally vertex-color-avoiding connectivity.

Now for simplicity, we also define some specific subgraphs of an edge- or vertex-colored graph which we will use during our proofs.

Definition 2.5. Given an edge-colored graph $G = (V, E)$ with color set $C = \{c_1, ..., c_k\}$, let us denote by $G_{\overline{c_i}}$ the graph which can be obtained from G by removing all the edges of color c_i.

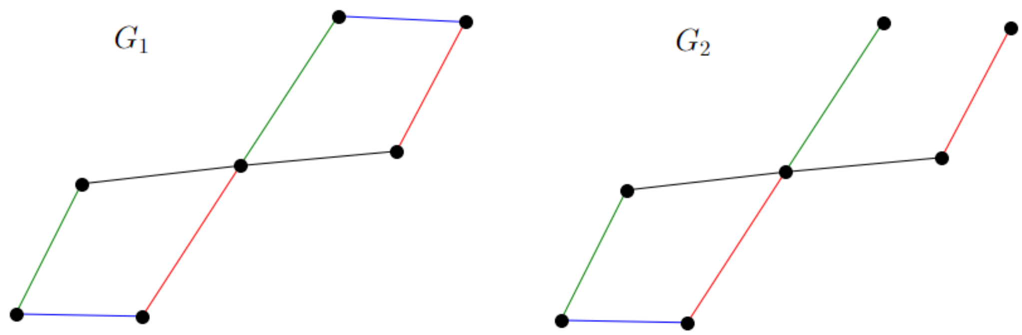


Figure 1: Examples on edge-color-avoiding connectivity: the graph G_1 is edge-color-avoiding connected, while the graph G_2 is not.

Given a vertex-colored graph $G = (V, E)$ with vertex set $V = \{v_1, ..., v_n\}$ and color set $C = \{c_1, ..., c_k\}$, let us denote by $G_{\overline{c_i}}$ the graph which can be obtained from G by removing the vertices of color c_i. Also we denote by $G_{\overline{c_i},v_j,v_l}$ the graph which can be obtained from G by removing the vertices of color c_i except the vertices v_j and v_l.

Although the same notation is given for the cases of edge-, vertex- and internally vertex-color-avoiding connectivity, this do not cause any misunderstanding since we always consider only one type of color-avoiding connectivity at once.

In Figures 3 and 4 we can see examples for the graphs defined in Definition 3.5.

After these definitions let us look at some simple properties of this concept.

Claim 2.1. *Let $G = (V, E)$ be a graph and $c : E \to C$ a function, which assigns color to the edges. Let $R_{ECA} \subseteq V \times V$, such that $(x, y) \in R_{ECA}$ if and only if x and y are edge-color-avoiding connected for any $(x, y) \in V \times V$. Then R_{ECA} defines an equivalence relation on the set of vertices.*

Proof. For any color $c_i \in C$, let $R_{\mathrm{ECA},c_i} \subseteq V \times V$ such that $(x, y) \in R_{\mathrm{ECA},c_i}$ if and only if x and y are edge-color-avoiding connected for any $(x, y) \in V \times V$.

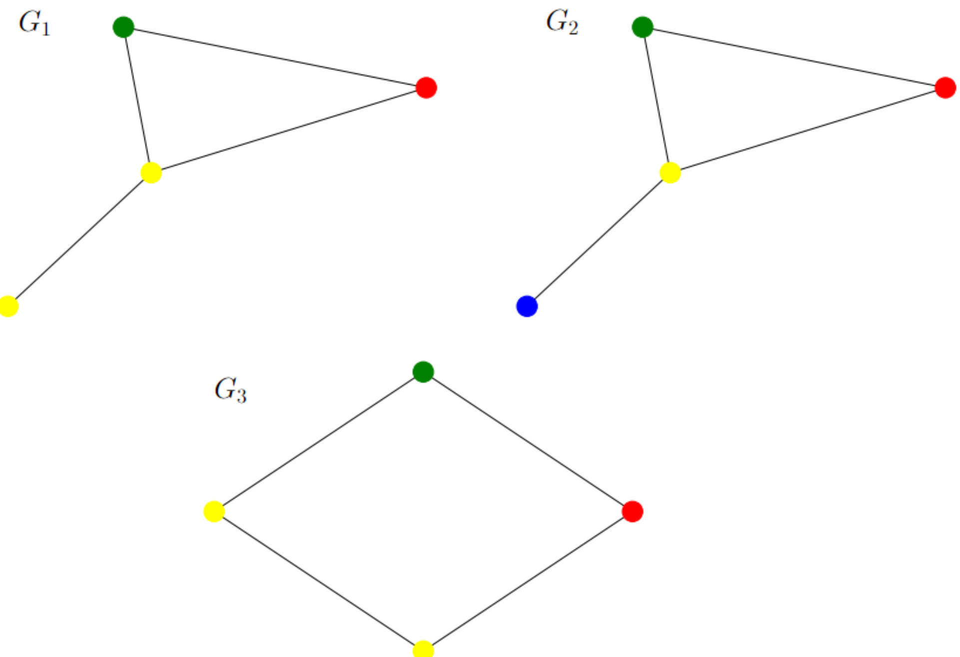


Figure 2: Examples on vertex- and internally vertex-color-avoiding connectivity: the graph G_1 is vertex-color-avoiding connected but not internally vertex-color-avoiding connected, the graph G_2 is neither vertex- nor internally vertex-color-avoiding connected and G_3 is both vertex- and internally vertex-color-avoiding connected.

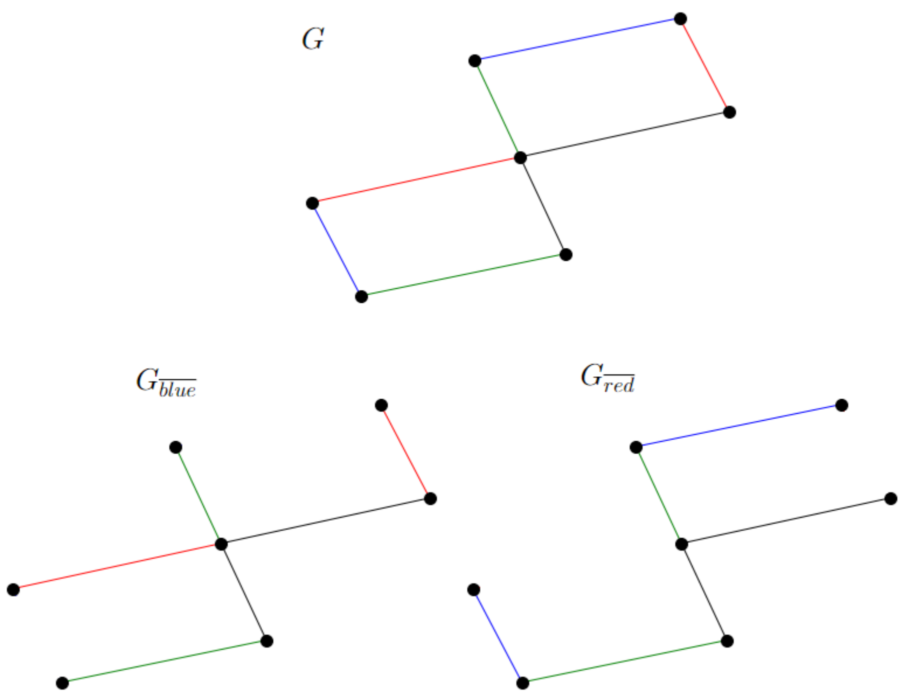


Figure 3: Examples on the graphs defined in Definition 2.5: for the edge-colored graph G, we can see the graphs $G_{\overline{\text{red}}}$ and $G_{\overline{\text{blue}}}$, which can be obtained from G by removing the red and the blue edges, respectively.

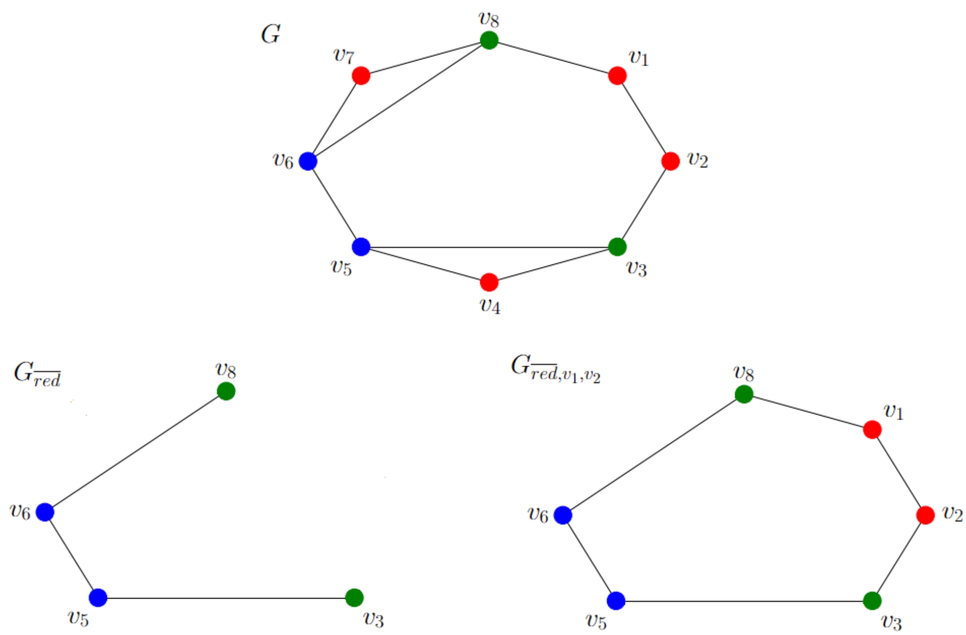


Figure 4: Examples on the graphs defined in Definition 2.5: for the vertex-colored graph G, we can see the graphs $G_{\overline{\text{red}}}$ and $G_{\overline{\text{red}},v_1,v_2}$. The former one can be obtained from G by removing all the red vertices, but in the latter case we do not remove vertices v_1 and v_2.

It easy to see that R_{ECA,c_i} is an equivalence relation since it can be considered as the usual (uncolored) connectivity relation in $G_{\overline{c_i}}$. By the definition of edge-color-avoiding connectivity, $R_{\text{ECA}} = \cap_{c_i \in C} R_{\text{ECA},c_i}$ holds, i.e., R_{ECA} can be expressed as an intersection of equivalence relations, thus R_{ECA} is also an equivalence relation. $\square$

Claim 2.2. *For a vertex-colored graph $G = (V, E)$, let $R_{VCA}, R_{IVCA} \subseteq V \times V$ such that $(x, y) \in R_{VCA}$ if and only if x and y are vertex-color-avoiding connected, and $(x, y) \in R_{IVCA}$ if and only if x and y are internally color-avoiding connected for any $(x, y) \in V \times V$. Then there exists a graph G for* *which $R_{VCA} \neq R_{IVCA}$, and neither R_{IVCA} nor R_{VCA} is an equivalence relation on the set of vertices.*

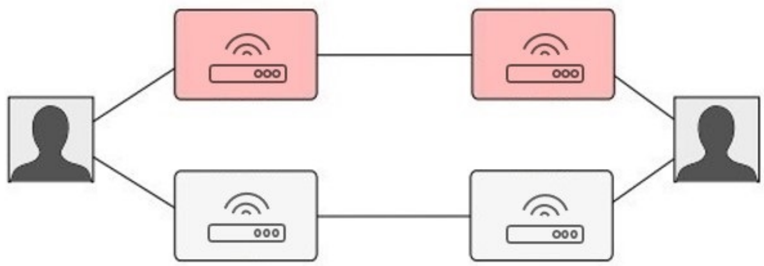
Proof. In Figure 5 we can see a simple example for a graph G for which $R_{\text{VCA}} \neq R_{\text{IVCA}}$ and neither R_{VCA} nor R_{IVCA} is an equivalence relation.

Figure 5: An example for Claim 2.2. The vertices v_1 and v_3 are not internally vertex-color-avoiding connected since they are not internally vertex-yellow-avoiding connected. Nonetheless, v_1 and v_3 are vertex-color-avoiding connected since v_1 and v_2 have the same color. Thus $R_{\text{VCA}} \neq R_{\text{IVCA}}$. Furthermore, v_2 and v_3 are vertex- and internally vertex-color-avoiding connected since they are connected by an edge, and so are v_3 and v_4. But v_2 and v_4 are neither vertex- nor internally vertex-color-avoiding connected thus transitivity does not hold, which means that neither R_{VCA} nor R_{IVCA} is an equivalence relation.

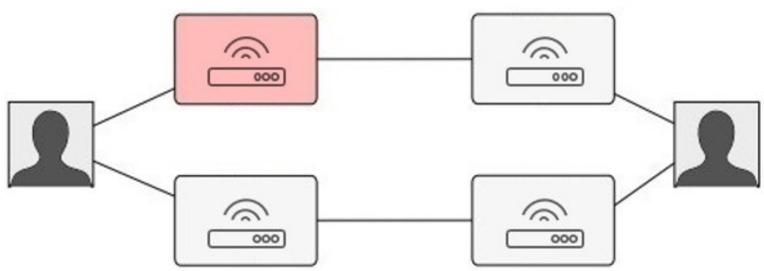
□

2.2 Previous results

In this section we provide a summary of previous results on color-avoiding connectivity. The concept was introduced by Krause et al. [11] with the motivation to develop a framework which can treat the heterogeneity of multiple vulnerable classes of vertices. In the paper they demonstrated how the heterogeneity of vertex classes can be exploited to maintain functionality of a complex network by utilizing multiple paths, mostly on communication networks; for one of their examples see Figure 6.



both avoidable



white not avoidable

Figure 6: An example on the motivation of color-avoiding connectivity about avoidable and nonavoidable adversaries. In this scenario two persons are communicating via the given networks where the network providers (represented by white and red vertices) can intercept the messages. Thus for safety reasons, the messages are divided into two parts for safety reasons and sent through different paths to the receiver. In the upper network, the network providers can only intercept half of the message, so both adversaries are avoidable. In the bottom network, the white network provider is not avoidable, they can intercept the whole message. *Source: Krause et al.* [11]

Krause et al. [11] provide an analytic framework for the internally vertex-color-avoiding percolation of random networks and also present its applicability on the AS-level Internet. The AS-level network is a collection of connected Internet protocol (IP) routing prefixes under the control of one or more network operators on behalf of a single administrative entity or domain. They colored the vertices of the network according to in which country the corresponding router is registered and supposed that every country is eavesdropping on the traffic but no countries share information. The last condition does not restrict generality since if two countries cooperated, then we could just color their vertices with the same color. In this colored graph if two routers are in the same internally vertex-color-avoiding connected component, then the communication between those two routers is safe since the messages can be divided and sent through different paths such that no country can intercept the whole message. They found that 26228 out of 49743 (roughly 53%) of the routers are in the largest color-avoiding connected component, but the results greatly differ from country to country. For instance, only 25% of the routers registered to the USA are in the largest color-avoiding connected component compared to 89% of the routers registered to Russia.

Krause et al. extended this original theory in [12]. They analyzed the effect of the frequency of vertices of different colors on the robustness of the networks. For unequal color frequencies, they found that the colors with the largest frequencies control vastly the robustness of the network, and colors of small frequency only plays a little role. In Krause et al. [9] the color-avoiding connectivity was further extended from vertex-colored graphs to edge-colored graphs. The researchers provided some important percolation theoretical results about the robustness of scale-free networks with respect to the color-avoiding connectivity. Giusfredi and Bagnoli investigated color-avoiding percolation in diluted lattices [6] and also showed that color-avoiding connectivity can be formulated as a self-organized critical problem, in which the asymptotic phase space can be obtained in one simulation [5]. Kryven established a generic analytic theory that describes how structure and sizes of all color-avoiding connected components in a network are affected by simple and color-dependent bond percolation. His theory predicts locations of the phase transitions, existence of wide critical regimes that do not vanish in the thermodynamic limit, and a phenomenon of color-switching in small components [13].

Molontay and Varga [14] has investigated the computational complexity of finding the color-avoiding components of a graph. They considered the following problems:

EdgeColorAvoidingConnectedComponent
Instance: a graph G, a finite color set C, a function $c : E(G) \to C$, and a positive integer l.
Question: is it true that G has an edge-color-avoiding connected component of size at least l?

VertexColorAvoidingConnectedComponent
Instance: a graph G, a finite color set C, a function $c : V(G) \to C$, and a positive integer l.
Question: is it true that G has a vertex-color-avoiding connected component of size at least l?

InternallyVertexColorAvoidingConnectedComponent
Instance: a graph G, a finite color set C, a function $c : V(G) \to C$, and a positive integer l.
Question: is it true that G has an internally vertex-color-avoiding connected component of size at least l?

The following theorems states the complexity of the problems.

Theorem 2.2.1 ([14])**.** *The problems* EdgeColorAvoidingConnectedComponent *and* VertexColorAvoidingConnectedComponent *are in P. and the problem* InternallyVertexColorAvoidingConnectedComponent *is NP-complete.*

Molontay and Varga [14] have also generalized the concept of color-avoiding connectivity by making the vertices or edges more vulnerable by assigning a list of colors to the vertices or edges representing all the vulnerabilities they have.

3 Some extremal problems of connectivity

In this section we collect some useful definitions and results which we make great use of in the case of color-avoiding connectivity. We recall some well-known concepts describing the robustness of an uncolored network like higher connectivity.

Definition 3.1. Let $G = (V, E)$ be a graph and k a positive integer. If the removal of any at most $k-1$ edges does not disconnect the graph, then G is called ***k*-edge-connected**. If the graph has at least $k+1$ vertices and the removal of any at most $k-1$ vertices does not disconnect the graph, then G is called ***k*-vertex-connected**.

In other words, a k-edge- or a k-vertex-connected graph can withstand an attack targeting at most $k-1$ edges or $k-1$ vertices, respectively.

To study extremal color-avoiding connected graphs, first we review some results on extremal uncolored, k-edge- or k-vertex-connected graphs. First, we consider the problem of determining the minimum number of edges of a k-edge- or k-vertex-connected graph on a given number of vertices. The solutions to these kinds of problems consist of two parts: giving a lower bound on the number of edges and showing that a graph can be constructed with that number of edges.

In the case of (1-vertex-)connectivity, the minimum number of edges is $n-1$ which is achieved by any tree on n vertices. For $k \geq 2$, it is known that the minimum number of edges of a k-vertex-connected graph on n vertices is $\lceil \frac{kn}{2} \rceil$ and a famous construction which achieves the lower bound is the $H_{k,n}$ Harary graph [7]; for some small Harary graphs see Figure 7. The proof of the correctness of this lower bound is based on the observation that every vertex of a k-vertex-connected graph has a degree of at least k since the removal of all the neighbors of any vertex leaves the vertex isolated.

Since a k-vertex-connected graph is k-edge-connected as well, the same lower bound can be given on the number of edges of any k-edge-connected

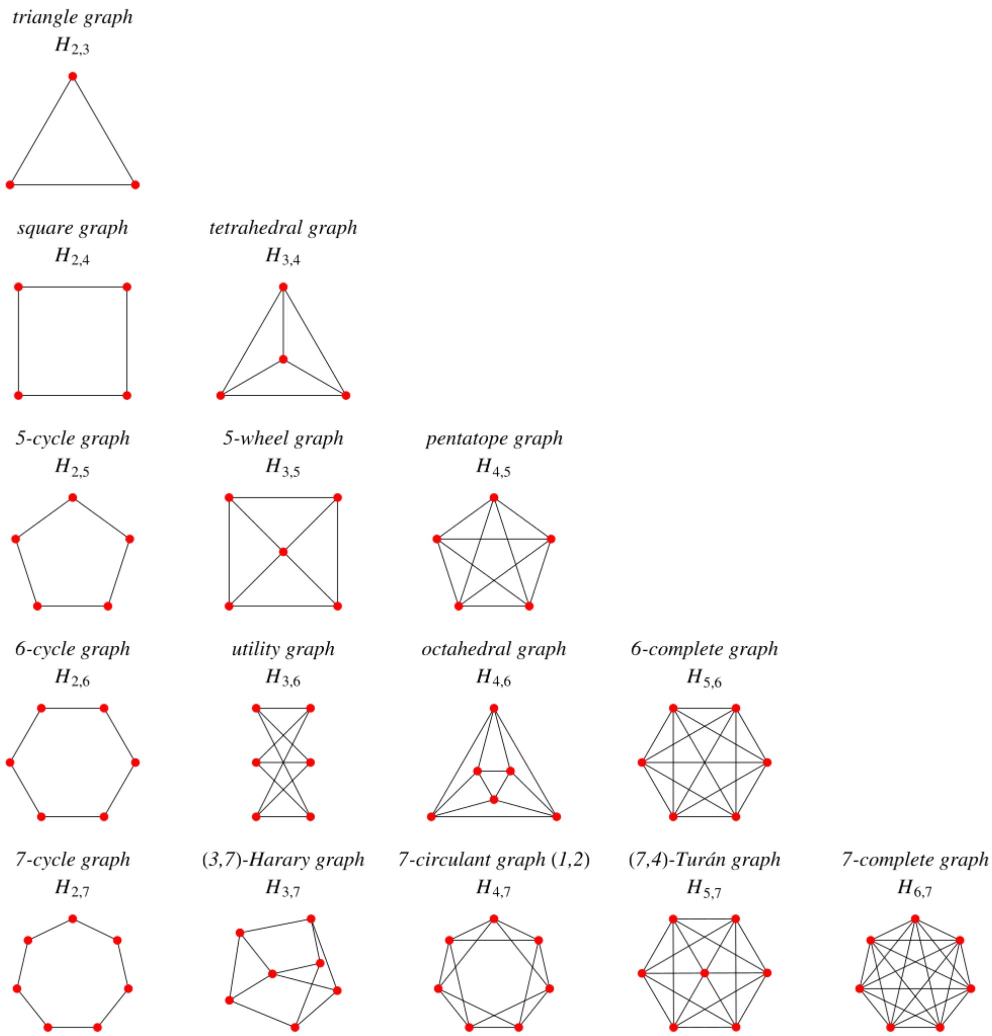


Figure 7: Some examples of small Harary graphs.
Source: Wolfram Mathworld, Harary Graph

graph.

The second problem we study is the following. Given a k-vertex-connected or a k-edge-connected graph, determine the maximum number of edges which can be removed so that the graph remains k-vertex-connected or k-edge-connected, respectively, or equivalently, find a k-edge- or k-vertex-connected spanning subgraph of the graph with minimum number of edges. The case $k = 1$ can be simply solved in polynomial time by finding a spanning tree for example with breadth-first search. However, this problem is NP-hard when $k \geq 2$ [1]. Now we explain why the case $k = 2$ is NP-hard since we use this statement later for proving the NP-hardness of the corresponding problem of the color-avoiding version.

As we saw, a 2-vertex- or 2-edge-connected graph has at least $\frac{2n}{2} = n$ edges, and if a 2-vertex- or 2-edge-connected graph has n edges, then the graph is a cycle. It is well-known that deciding whether a given graph contains a Hamiltonian cycle is NP-complete [4], so this problem is also NP-hard.

However, the problem of finding a 2-edge-connected spanning subgraph with minimum number of edges is approximable in polynomial time. The best known approximation factor for this problem is $\frac{4}{3}$ [8]. With some additional conditions on the input graph, there exist algorithms with even better approximation ratios [3, 15, 17]. Now we present the main idea of a polynomial time algorithm with an approximation ratio of $\frac{3}{2}$, namely the Khuller–Vishkin algorithm. The idea is to modify the depth-first search algorithm such that it does not just find a spanning tree, but a minimally 2-edge-connected spanning subgraph by the addition of extra edges, then achieving the minimality by the deletion of surplus edges.

4 Extremal color-avoiding connected graphs

In the next two sections we consider two extremal problems of color-avoiding connectivity, which naturally arise for each version of color-avoiding connectivity (edge, vertex and internally vertex):

- Determine the minimum number of edges in an edge-, vertex- or internally vertex-color-avoiding connected graph, where the edges or vertices are colored with exactly k colors.
- Determine the maximum number of edges that can be removed from an edge-, vertex- or internally vertex-color-avoiding connected graph so that the graph remains color-avoiding connected.

In the first scenario we need to build a whole color-avoiding connected network. In the second scenario we need to reduce the number of edges of a given network which is still color-avoiding connected. In real-life situations, this can be imagined as constructing a color-avoiding connected network with minimum construction costs or reducing the maintenance costs of a network while preserving its color-avoiding connectivity. Undeniably these examples are a lot less complex than in real life, but they give a good starting point, and also the negative consequences apply to real-life scenarios as well. For example if a color-avoiding connected graph cannot be constructed with less than a given number of edges, then it gives a lower bound on the construction costs as well no matter how many other constraints are given.

4.1 Edge-color-avoiding connectivity

After this brief introduction to this topic and its importance, we are going to present some of our results. First, we determine the minimum number of edges of an edge-color-avoiding connected graph with n vertices and k colors.

Theorem 4.1.1. *The minimum number of edges in an edge-color-avoiding connected graph on n vertices, where the edges are colored with exactly $k \geq 2$*

colors, is $n-1+\left\lceil\frac{n-1}{k-1}\right\rceil$*. In the case of* $k=1$*, there exist no edge-color-avoiding connected graphs on* $n \geq 2$ *vertices.*

Proof. First let us look at the case $k=1$. Suppose to the contrary that there exists an edge-color-avoiding graph on $n \geq 2$ vertices. Then by the definition of edge-color-avoiding connectivity, the graph must be connected, thus it must have edges. However, if every edge has the same color, then the removal of this single color clearly disconnects the graph, which is a contradiction.

Now consider the case $k \geq 2$. Let G be an edge-color-avoiding connected graph on n vertices, where the edges are colored with exactly k colors. Let $C=\{c_1,...,c_k\}$ be the set of colors of G. For any $i \in [k]$, the graph $G_{\overline{c_i}}$ must be connected, thus it must have at least $n-1$ edges. That sums up to $k \cdot (n-1)$ edges, where every edge is counted $k-1$ times, so the number of edges is at least $\left\lceil\frac{k\cdot(n-1)}{k-1}\right\rceil = n-1+\left\lceil\frac{n-1}{k-1}\right\rceil$.

To show that this lower bound is tight, consider the following edge-colored graph $G=(V,E)$. (For an example see Figure 8.)

- Let $V=\{v_0,...,v_{k-1}\}$ and $C=\{0,1,...,k-1\}$.
- Let $v_iv_{i+1} \in E$ and $c(v_iv_{i+1}) = (i \mod (k-1))$ for every $i \in \{1,...,n-1\}$, where c is the function which assigns colors to the vertices.
- Let $v_iv_{i+k-1} \in E$ and $c(v_iv_{i+k-1}) = k-1$ for every $i \in \{0,k-1,2k-2,...,\left\lfloor\frac{n-1}{k-1}\right\rfloor k-1\}$ and if $k-1$ does not divide $n-1$, then let $v_{\left\lfloor\frac{n-1}{k-1}\right\rfloor k-1}v_n \in E$ and $C(v_{\left\lfloor\frac{n-1}{k-1}\right\rfloor k-1}v_n) = k-1$.

It is not difficult to show that G is edge-color-avoiding connected and has $n-1+\left\lceil\frac{n-1}{k-1}\right\rceil$ edges. □

4.2 Vertex-color-avoiding connectivity

Now let us look at the vertex-color-avoiding connectivity. The following theorem gives a tight lower bound on the number of edges of a vertex-color-avoiding connected graph.

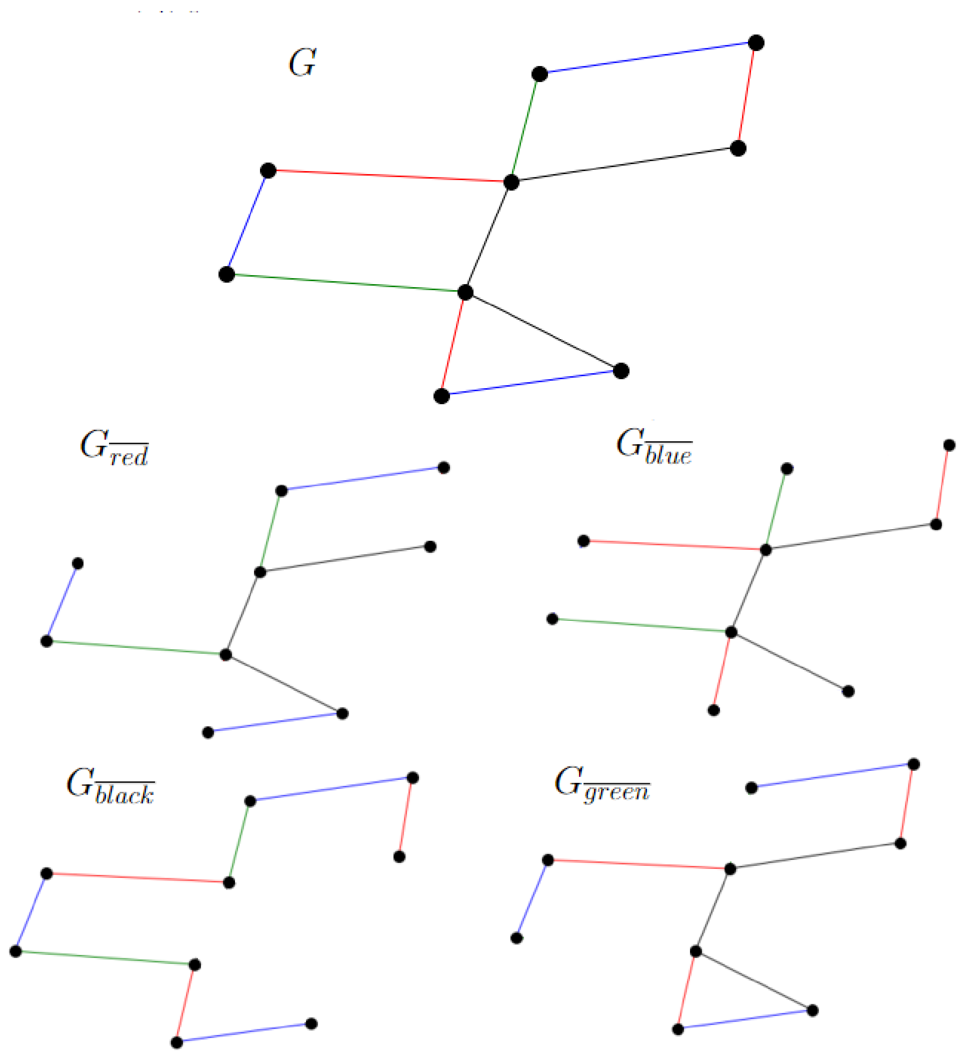


Figure 8: An edge-color-avoiding connected graph G on 9 vertices colored with exactly 4 colors and with minimum number of edges, and the graphs $G_{\overline{\text{red}}}$, $G_{\overline{\text{blue}}}$, $G_{\overline{\text{black}}}$ and $G_{\overline{\text{green}}}$.

Theorem 4.2.1. *The minimum number of edges in a vertex-color-avoiding connected graph on n vertices, where the edges are colored with exactly k ($k \geq 2$) colors, is n if $k \geq 3$ and $n-1$ if $k = 1$ or $k = 2$.*

For the proof of this theorem we are going to use the following lemma.

Lemma 1. *If v is a cut-vertex in a vertex-color-avoiding connected graph G, then all but one component of $G - v$ consist only of vertices of the color of v.*

Proof. Let G be a vertex-color-avoiding connected graph with a cut-vertex v of color c. Suppose to contrary that there are at least two components in $G - v$ which contain vertices of color different from c. Let v_1 and v_2 two vertices whose color is different than c and they are in different components in $G - v$. Then v_1 and v_2 are not vertex-c-avoiding connected. Thus it is a contradiction, which means that at most one component can contain vertices of different color from the color of the cut-vertex. □

Proof of Theorem 4.2.1. First let $k = 1$. In this case every vertex is of the same color, which means that by the definition of vertex-color-avoiding connectivity, every connected graph is color-avoiding connected. So the graph just has to be connected, which means that it must have at least $n-1$ edges, and clearly any tree on n vertices achieves this lower bound.

Now let $k = 2$. By the definition of vertex-color-avoiding connectivity, G must be connected, so it must have at least $n-1$ edges. Now we construct a vertex-color-avoiding connected graph with exactly $n-1$ edges. Let us take an arbitrary tree T and let e be an arbitrary edge of T. The removal of e would disconnect T into 2 components: let us color the vertices of one component with blue and the other one with red. With this vertex coloring the tree T is clearly a vertex-color-avoiding connected graph with $n-1$ edges.

Now let us look at the case of $k \geq 3$. First we will show that $n-1$ edges is not enough. Suppose to the contrary that there exists a vertex-color-avoiding connected graph G which has n vertices and $n-1$ edges and whose vertices are colored with exactly $k \geq 3$ colors. Since a vertex-color-avoiding connected graph must be connected, G has to be a tree. If the non-leaf vertices are all of color c, then there exist at least two leaves u and v with different color from c. Then u and v are not vertex-c-avoiding connected, which contradicts the definition of vertex-color-avoiding connectivity. So there exist two adjacent non-leaf vertices v_1 and v_2 of colors c_1 and c_2 ($c_1 \neq c_2$), respectively. By Lemma 1, since v_1 is a cut-vertex, all of the components of $G - v_1$ but one contain vertices only of color c_1. The remaining one component must be the one which contains v_2. The same can be told about v_2 and c_2. It is easy to see (see Figure 9) that by the adjacency of v_1 and v_2 this means that all of the vertices should be of color c_1 or c_2. Which contradicts the fact that the graph is colored with exactly $k \geq 3$ colors. Thus any vertex-color-avoiding connected graph on n vertices and with exactly $k \geq 3$ colors must have at least n edges.

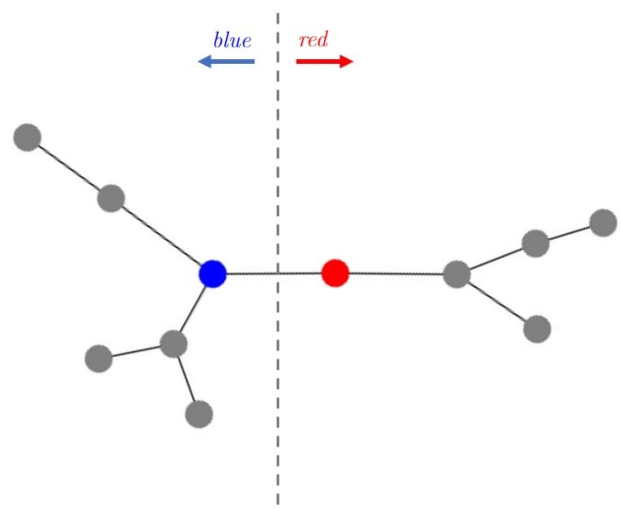


Figure 9: A vertex-color-avoiding tree with two adjacent non-leaf vertices of different colors. By Lemma 1, the vertices which are on the left side of the dashed line must be blue and the vertices on the right side must be red.

Now we construct a vertex-color-avoiding graph G with n vertices, n edges and exactly $k \geq 3$ colors. For an example see Figure 10.

- Let $V = \{v_{0,0}, v_{0,1}, ..., v_{0,n_0}, v_{1,0}, ..., v_{1,n_1}, ..., v_{k-1,0}, ..., v_{k-1,n_{k-1}}\}$, where n_i is the number of vertices of color i and $C = \{0, ..., k-1\}$.
- Let $(v_{i,j}, v_{i,j+1}) \in E$ for every $i \in \{0, 1, ..., n_j\}$, for every $j \in C$.
- Let $(v_{j,n_j}, v_{j+1,0}) \in E$ for every $j \in \{0, .., k-2\}$ and let $(v_{k-1,n_{k-1}}, v_{0,0}) \in E$.

It is not difficult to show that this graph is vertex-color-avoiding connected and has n edges.

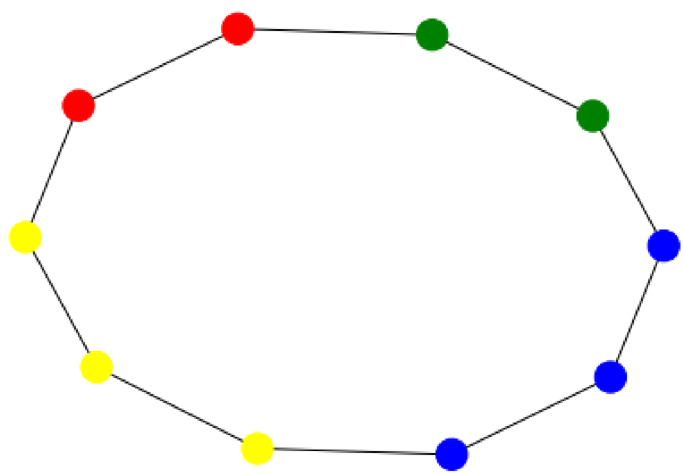


Figure 10: A vertex-color-avoiding connected graph G on 10 vertices colored with exactly 4 colors and with minimum number of edges.

$\square$

4.3 Internally vertex-color-avoiding connectivity

In this section we give a tight lower bound on the number of edges of an internally vertex-color-avoiding connected graph.

Theorem 4.3.1. *The minimum number of edges in an internally vertex-color-avoiding connected graph on n vertices, where the edges are colored with exactly $k \geq 2$) colors, is $\left\lceil \frac{2k-1}{2k-2}n - \frac{k}{k-1} \right\rceil$, and if $k = 1$, then it is $\binom{n}{2}$.*

Proof. First let us look at the case $k = 1$. It is not difficult to see that only the complete graph is internally color-avoiding connected, so the minimum number of edges of an internally color-avoiding graph on n vertices, where the vertices are colored with the same color, is indeed $\binom{n}{2}$.

Now let us look at the case $k \geq 2$. Let $G = (V, E)$ be an internally vertex-color-avoiding connected graph on n vertices, whose vertices are colored with colors $C = \{c_1, ..., c_k\}$ and each color of C is used at least once. Let x be the number of edges whose endvertices are of the same color, and let n_i be the number of vertices of color c_i. We are going to prove two lower bounds on the number of edges using x, then combine the two bounds to get rid of this variable.

Let us prove the first bound, namely, $(k-2)|E| \geq (k-1)n - k - x$. By color-avoiding connectivity, for any $i \in \{1, ..., k\}$ the graph $G_{\overline{c_i}}$ is connected, so it must have at least $n - n_i - 1$ edges. Altogether, we counted

$$\sum_{i=1}^{k} n - n_i - 1 = kn - n - k = (k-1)n - k$$

edges, but the edges whose endpoints have the same color were counted $k - 1$ times, and the other edges (i.e. whose endpoints are not of the same color) were counted $k - 2$ times. This means that $(k-2)|E| \geq (k-1)n - k - x$, which is the same inequality as the one stated above.

Now let us prove the second bound, namely $k|E| \geq kn - k + x$. For any $c_i \in C$ by the definition of internally vertex-c_i-color-avoiding connectivity, every vertex of color c_i should have a neighbour of a different color than c_i. Thus removing the edges whose both endpoints are of color c_i leaves the graph connected, so the remaining graph should have at least $n-1$ edges. If we count the remaining edges for every $c_i \in C$, then altogether we counted $k(n-1)$ edges, but the edges whose endpoints have different colors were counted k times, and the other edges were counted $k-1$ times. This means that $ke - x \geq k(n-1)$, which is the same inequality as the one stated above.

Adding these two inequalities together, we obtain

$$(2k-2)|E| \geq (2k-1)n - 2k$$

which proves the lower bound part of the theorem.

Now we show that this lower bound is tight, i.e. we construct an internally vertex-color-avoiding connected graph on n vertices with exactly k colors and $\lceil \frac{2k-1}{2k-2}n - \frac{k}{k-1} \rceil$ edges. First we consider the special case when the number of vertices is of the form $n = 2(k-1)m + 2$ for some $m \in \mathbb{N}$. In this case the lower bound is

$$\left\lceil \frac{2k-1}{2k-2}n - \frac{k}{k-1} \right\rceil = \left\lceil \frac{2k-1}{2k-2}(2(k-1)m+2) - \frac{k}{k-1} \right\rceil =$$

$$= (2k-1)m + \left\lceil \frac{4k-2}{2k-2} - \frac{2k}{2k-2} \right\rceil = (2k-1)m + 1,$$

and the following graph is internally vertex-color-avoiding connected and has $(2k-1)m+1$ edges. (For an example see Figure 11.):

- Let $C = \{1, ..., k\}$ and $V = \Big\{v_{1,j}, v_{k,j} \Big| j \in \{1, \ldots, m+1\}\Big\} \cup \Big\{v_{i,j}, v'_{i,j} \Big| i \in \{2, ..., k-1\},\ j \in \{1, ..., m\}\Big\}$ where $v_{i,j}$ and $v'_{i,j}$ have color i for any $i \in \{1, ..., k\}$ and $j \in \{1, ..., m+1\}$.

- Let $v_{i,j}v_{i,j+1} \in E$ for any $i \in \{1, ..., m\}$ and $j \in \{1, k\}$.

- Let $v_{i,j}v'_{i,j} \in E$ and $v'_{i,j}v_{i+1,j} \in E$ for any $i \in \{1, ..., k-1\}$ and $j \in$

$\{1, ..., m\}$ and also let $v_{1,m+1}v_{k,m+1} \in E$.

It is easy to see that this graph is internally vertex-color-avoiding connected and has $(2k-1)m+1$ edges.

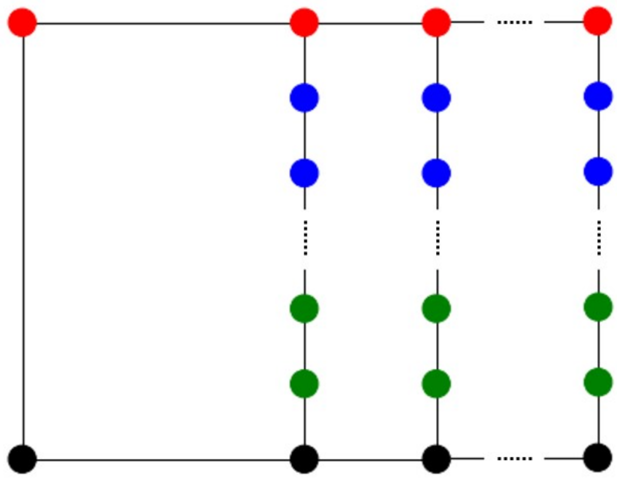


Figure 11: An internally vertex-color-avoiding connected graph G on n vertices colored with exactly k colors and with minimum number of edges.

Now we construct a graph $G = (V, E)$ for any n with exactly k colors and minimum number of edges. Clearly, any positive integer n can be written in the form $n = 2(k-1)m + 2 - l$, where $1 \leq l \leq 2(k-1) - 1$. Based on this form of n consider the following vertex-colored graph $G = (V, E)$ (for examples see Figure 12):

- Take the previous construction on $2(k-1)m + 2$ vertices.
- If l is odd and $l \leq 2(k-1) - 2$, then remove the vertices $\{v_{2,m}, v'_{2,m}, ..., v_{2+\frac{l-1}{2},m}\}$. If l is even and $l \leq 2(k-1) - 2$, then remove the vertices $\{v_{2,m}, v'_{2,m}, ..., v_{2+\frac{l-2}{2},m}, v'_{2+\frac{l-2}{2},m}\}$. If $l = 2(k-1) - 1$, then remove the vertices $\{v_{1,m+1}, v_{2,m}, v'_{2,m}, ..., v_{k-1,m}, v'_{k-1,m}\}$. Connect the two neighbours of the removed vertices.

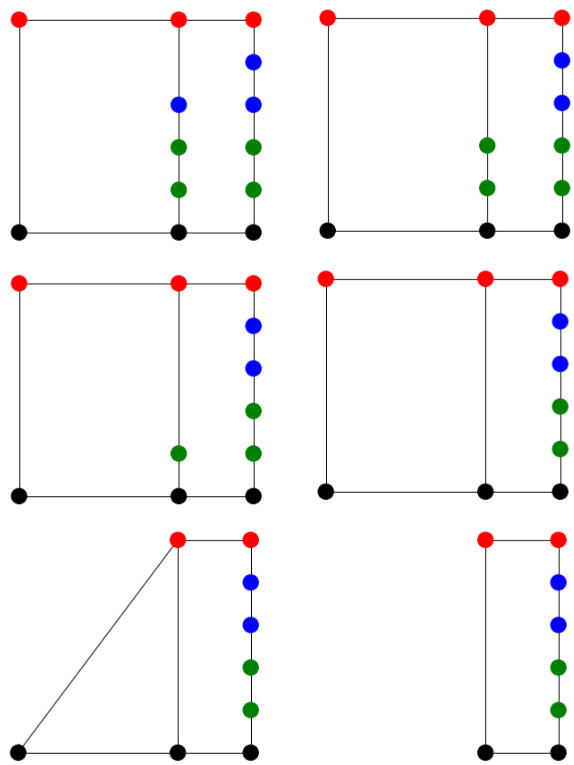


Figure 12: Some internally vertex color-avoiding connected graphs on $n = 13, 12, 11, 10, 9, 8$ vertices colored with exactly $k = 4$ colors with minimum number of edges.

It is easy to see that this graph is still internally vertex-color-avoiding connected and now we show that it has minimum number of edges.

The given lower bound is:

$$\left\lceil \frac{2k-1}{2k-2}((2k-2)m+2-l) - \frac{k}{k-1} \right\rceil = (2k-1)m+1-l\left\lfloor \frac{2k-1}{2k-2} \right\rfloor$$
$$= (2k-1)m+1-l-\left\lfloor \frac{l}{2k-2} \right\rfloor,$$

and it is not difficult to see that the graph G has exactly this many edges. $\square$

5 Color-avoiding connected spanning subgraphs with minimum number of edges

5.1 Complexity results

In this section we study the problem of determining the maximum number of edges which can be removed so that the graph remains color-avoiding connected. Here, we investigate the complexity the following problems.

ECA-CSS (short for Edge-Color-Avoiding Connected Spanning Subgraph)
Instance: a graph G, a finite color set C, a function $c : E(G) \to C$, and a positive integer l.
Question: is it true that G has an edge-color-avoiding connected spanning subgraph with at most l edges?

VCA-CSS (short for Vertex-Color-Avoiding Connected Spanning Subgraph)
Instance: a graph G, a finite color set C, a function $c : V(G) \to C$, and a positive integer l.
Question: is it true that G has a vertex-color-avoiding connected spanning subgraph with at most l edges?

IVCA-CSS (short for Internally Vertex-Color-Avoiding Connected Spanning Subgraph)
Instance: a graph G, a finite color set C, a function $c : V(G) \to C$, and a positive integer l.
Question: is it true that G has an internally vertex-color-avoiding connected spanning subgraph with at most l edges?

Now we present a theorem about the complexity of the problems.

Theorem 5.1.1. *The problems ECA-CSS, VCA-CSS and IVCA-CSS are NP-complete.*

Proof. First, note that the problems ECA-CSS, VCA-CSS and IVCA-CSS are in NP since a witness for each of these problems is an edge-, vertex- and internally vertex-color-avoiding-connected spanning subgraph with at most l edges.

To show that ECA-CSS is NP-hard, note that if every edge has different color, then the graph is edge-color-avoiding connected if and only if it is 2-edge-connected, so in this case the problem ECA-CSS is equivalent to the Hamiltonian cycle problem, which is known to be NP-complete [4].

Now let us look at the problems VCA-CSS and IVCA-CSS. First notice that if every vertex has different color, then the graph is vertex-color-avoiding connected if and only if it is internally vertex-color-avoiding connected, in addition, these are equivalent to the uncolored graph being 2-vertex-connected. So in this case the problems VCA-CSS and IVCA-CSS are equivalent to the Hamiltonian cycle problem, which is known to be NP-complete [4]. $\square$

5.2 Approximation algorithms

In this section we present polynomial time approximation algorithms for the problems of finding an edge-, vertex- or internally vertex-color-avoiding connected spanning subgraph with minimum number of edges. The main idea of these algorithms is similar to the Khuller–Vishkin algorithm [10]. First we find a spanning tree, then for each color we select some additional edges to obtain a color-avoiding-connected subgraph, and finally we remove the surplus edges.

To shorten the description of the algorithms we introduce the following simple subroutines.

SpanningTree
Input: a connected graph G.
Output: a spanning tree of G.

ConnectedComponents

Input: a graph G.

Output: the vertex sets of the connected components.

ContractVertices

Input: a graph G and a set H of disjoint sets of vertices of G.

Output: a graph which can be obtained from G by contracting the vertices of each set in H into a single vertex.

DifferentColor

Input: a graph G and a vertex v.

Output: True if v has a neighbour of a different color than its own color, otherwise False.

DifferentColorEdge

Input: a graph G in which every vertex has a neighbor with a different color than its own color and a vertex v.

Output: an edge which connects v to a vertex which has a different color than the color of v.

First, consider the following algorithm for finding an edge-color-avoiding connected spanning subgraph.

Algorithm 1 Finding an edge-color-avoiding connected spanning subgraph

---

Input: An edge-color-avoiding connected graph G with color set C

Output: An edge-color-avoiding connected spanning subgraph $\hat{G}$ of G

```
  Ĝ ← SpanningTree(G)
  for c ∈ C do
      if Ĝ_c̄ is not connected then
          H′ ← ConnectedComponents(Ĝ_c̄)
          G′ ← ContractVertices(G_c̄, H′)
          E(Ĝ) ← E(Ĝ) ∪ E(SpanningTree(G′))
  for e ∈ E do
      if Ĝ − e is edge-color-avoiding connected then
          Ĝ ← Ĝ − e
  return Ĝ
```

Now we prove that the above algorithm is correct and the found edge-color-avoiding spanning subgraph has at most $2(k-1)/k$ times as many edges as an optimal one, i.e. one with minimum number of edges.

Theorem 5.2.1. *Algorithm 1 is a polynomial time $\frac{2(k-1)}{k}$-approximation algorithm for the problem of finding an edge-color-avoiding connected spanning subgraph with minimum number of edges.*

Proof. Since G is edge-color-avoiding connected, it is connected as well, thus it has a spanning tree and in the first step the algorithm selects the edges of the spanning tree. Now we show that in the second phase the algorithm selects some additional edges to ensure edge-color-avoiding connectivity. By the edge-color-avoiding connectivity of G, the graph $G_{\overline{c}}$ is connected for any color c and any graph obtained from $G_{\overline{c}}$ by contracting some disjoint sets of vertices is also connected, thus it has a spanning tree. With the selection of the edges of these spanning trees the graph of the selected edges becomes edge-color-avoiding connected. In the third phase the algorithm removes some selected edges so that the graph of the selected edges remains edge-color-avoiding connected. Therefore, the algorithm finds an edge-color-avoiding connected spanning subgraph.

Now we prove that the found spanning subgraph has at most $2\frac{k-1}{k} \cdot \frac{k(n-1)}{k-1}$ edges, which is, by Theorem 4.1.1, at most $2\frac{k-1}{k}$ times as many as the minimum number of edges of an edge-color-avoiding connected graph whose edges are colored with exactly k colors, implying that Algorithm 1 is a $\frac{2k-1}{/}k$-approximation algorithm. In the first step the algorithm selects the edges of a spanning tree T, which clearly consists of $n-1$ edges. In the second step, for each color c, if the removal of the edges of color c disconnects the graph of the so far selected edges, the algorithm selects some additional edges of different color than c to achieve edge-c-avoiding connectivity in the graph of the selected edges: more precisely, if the removal of color c from the graph of the selected edges leaves m connected components, then the algorithm selects $m-1$ new edges. Let e_c denote the number of edges of color c in T

for any color c. Since T was selected in the first step, the removal of color c from the graph of the so far selected edges leaves at most $e_c + 1$ components. Therefore, the algorithm selects at most

$$(n-1) + \sum_{c \in C} e_c = 2(n-1)$$

edges since $\sum_{c \in C} e_c = n-1$, Finally, in the third phase, the algorithm removes some selected edges, thus the found subgraph at the end has indeed at most $2(n-1)$ edges. Also, if OPT denotes the minimum number of edges of an edge-color-avoiding connected graph then

$$2(n-1) = \frac{2(k-1)}{k} \cdot \frac{k(n-1)}{k-1} = \frac{2(k-1)}{k} \cdot \mathrm{OPT},$$

so the given algorithm is a $\frac{2(k-1)}{k}$-approximation algorithm.

Finally, we prove that the algorithm runs in polynomial time. The first step, which is the construction of a spanning tree, can be done in time $O(|V| + |E|)$ with for example breadth-first search. Then for a given color c we remove the edges of color c of the tree, which can be done in time $O(|E|)$. Then we contract some edges of this graph, which can be done in time $O(|V| + |E|)$ and the contraction part can be done in time $O(|E|)$. Then we also construct a spanning tree of the so obtained graph which can be done in time $O(|V| + |E|)$. We do these steps for every color $c \in C$ so this phase takes $O(|C| \cdot (|V| + |E|))$ time. The last step, where we check for every edge, whether it can be removed so that the so far selected subgraph remains edge-color-avoiding connected, that is, for every selected edge (which is at most $2(|V| - 1)$ edges) and for each color we check whether the removal of this edge and this color does not disconnect the found subgraph, which can be done in time $O(|C| \cdot |V|^2)$. Therefore, the running time of the algorithm is $O(|C||V|^2)$. $\square$

Now we present a similar algorithm for the case of vertex-color-avoiding connectivity.

Algorithm 2 Finding a vertex-color-avoiding connected spanning subgraph

Input: A vertex-color-avoiding connected graph G with color set C.
Output: A vertex-color-avoiding connected spanning subgraph $\hat{G}$ of G.

```
Ĝ ← SpanningTree(G)
for c ∈ C do
    if Ĝ_c̄ is not connected then
        H′ ← ConnectedComponents(Ĝ_c̄)
        G′ ← ContractVertices(G_c̄, H′)
        E(Ĝ) ← E(Ĝ) ∪ E(SpanningTree(G′))
for e ∈ E do
    if Ĝ − e is vertex-color-avoiding connected then
        Ĝ ← Ĝ − e
return Ĝ
```

Like in the case of edge-color-avoiding connectivity, we present a similar theorem about this algorithm.

Theorem 5.2.2. *Algorithm 2 is a polynomial time* 2*-approximation algorithm for the problem of finding a vertex-color-avoiding connected spanning subgraph with minimum number of edges.*

Proof. Since G is vertex-color-avoiding connected, it is connected as well, thus it has a spanning tree and in the first step the algorithm selects the edges of the spanning tree. Now we show that in the second phase the algorithm selects some additional edges to ensure vertex-color-avoiding connectivity. By the vertex-color-avoiding connectivity of G, the graph $G_{\overline{c}}$ is connected for any color c and any graph obtained from $G_{\overline{c}}$ by contracting some disjoint sets of vertices is also connected, thus it has a spanning tree. With the selection of the edges of these spanning trees the graph of the selected edges becomes vertex-color-avoiding connected. In the third phase the algorithm removes some selected edges so that the graph of the selected edges remains vertex-color-avoiding connected. Therefore, the algorithm finds a vertex-color-avoiding connected spanning subgraph.

Now we prove that the found spanning subgraph has at most $2n$ edges, which is, by Theorem 4.2.1, at most 2 times as many as the minimum number of edges of a vertex-color-avoiding connected graph whose vertices are colored with exactly k colors, implying that Algorithm 2 is a 2-approximation algorithm. In the first step the algorithm selects the edges of a spanning tree T, which clearly consists of $n-1$ edges. In the second phase, for each color c, if the removal of the vertices of color c disconnects the graph of the so far selected edges, the algorithm selects some additional edges to achieve vertex-c-avoiding connectivity in the graph of the selected edges: more precisely, if the removal of color c from the graph of the selected edges leaves m connected components, then the algorithm selects $m-1$ new edges. The removal of vertices of color c from the graph of the selected edges disconnects the graph into at most

$$1 + \sum_{\substack{v \in V: \\ c(v)=c}} (d_T(v) - 1)$$

components, where $c(v)$ is the color of v and $d_T(v)$ is the degree of v in T, since the removal of any vertex v creates at most $d_T(v) - 1$ new components. Therefore, the algorithm selects at most

$$\begin{aligned} n-1+\sum_{c \in C} \sum_{\substack{v \in V: \\ c(v)=c}} (d_T(v)-1) = (n-1)+\sum_{v \in V} d_T(v)-n = (n-1)+2(n-1)-n \\ = 2n-3 \end{aligned}$$

edges. Finally, in the third phase, the algorithm removes some selected edges, thus the found subgraph at the end has indeed at most $2n-3$ edges. Also, if OPT denotes the minimum number of edges of a vertex-color-avoiding connected graph then

$$2n - 3 \leq 2n = 2 \cdot \mathrm{OPT},$$

so the given algorithm is a 2-approximation algorithm.

Similarly as Algorithm 1, this algorithm also runs in time $O(|C||V|^2)$. $\square$

Now we present an algorithm for finding an internally vertex-color-avoiding connected spanning subgraph.

Algorithm 3 Finding an internally vertex-color-avoiding connected spanning subgraph

---

Input: An internally vertex-color-avoiding connected graph G with color set C.

Output: An internally vertex-color-avoiding connected spanning subgraph $\hat{G}$ of G.

$\hat{G} \leftarrow \text{SpanningTree}(G)$
for $c \in C$ **do**
  if $\hat{G}_{\overline{c}}$ is not connected **then**
    $H' \leftarrow \text{ConnectedComponents}(\hat{G}_{\overline{c}})$
    $G' \leftarrow \text{ContractVertices}(G_{\overline{c}}, H')$
    $E(\hat{G}) \leftarrow E(\hat{G}) \cup E(\text{SpanningTree}(G'))$
for $v \in V$ **do**
  if not $\text{DifferentColor}(\hat{G}, v)$ **then**
    $\hat{G} \leftarrow \hat{G} + \text{DifferentColorEdge}(G, v)$
for $e \in E$ **do**
  if $\hat{G} - e$ is vertex-color-avoiding connected **then**
    $\hat{G} \leftarrow \hat{G} - e$
return $\hat{G}$

As before, we also present a theorem about the algorithm.

Theorem 5.2.3. *Algorithm 3 is a polynomial time $3\frac{2k-2}{2k-1}$-approximation algorithm for the problem of finding an internally vertex-color-avoiding connected spanning subgraph with minimum number of edges.*

Proof. Similarly to Algorithm 2 the graph of the selected edges after the second phase is vertex-color-avoiding connected, but not necessarily internally vertex-color-avoiding connected: any vertex v might not be internally vertex-$c(v)$-avoiding connected to the other vertices, where $c(v)$ denotes the color of v. Now we show that the third phase of Algorithm 3 ensures internally vertex-color-avoiding connectivity. Let v and w be two arbitrary vertices of G. It is enough to show that after the third phase there exists an internally vertex-$c(v)$-avoiding path between v and w.

Case 1: w is not of color $c(v)$. In the third phase the algorithm selects an edge uv where the color of u is not $c(v)$. After the second phase u and w are vertex-$c(v)$-avoiding connected, and since neither u nor w are of color $c(v)$, there exists a vertex-$c(v)$-avoiding path between u and w in the graph of the so far selected edges. By the extension of this path with uv we obtain an internally vertex-$c(v)$-avoiding path between v and w.

Case 2: w is of color $c(v)$. In the third phase the algorithm selects two edges u_1v and u_2w where neither u_1 nor u_2 is of color $c(v)$. After the second phase u_1 and u_2 are vertex-$c(v)$-avoiding connected, and since neither u_1 nor u_2 are of color $c(v)$, there exists a vertex-$c(v)$-avoiding path between u and w in the graph of the so far selected edges. By the extension of this path with u_1v and u_2w we obtain an internally vertex-$c(v)$-avoiding path between v and w.

Hence the graph of the selected edges after the third phase is indeed internally vertex-color-avoiding connected. In the fourth phase the algorithm removes some selected edges so that the graph of the selected edges remains internally vertex-color-avoiding connected. Therefore, the algorithm indeed produces an internally vertex-color-avoiding connected spanning subgraph.

Now we prove that the spanning subgraph has at most $3\frac{2k-2}{2k-1}\left(\frac{2k-1}{2k-2}n-\frac{k}{k-1}\right)$ edges, which is, by Theorem 4.3.1, at most $3\frac{2k-2}{2k-1}$ times as many as the minimum number of edges of an internally vertex-color-avoiding connected graph whose vertices are colored with exactly k colors, implying that Algorithm 3 is a $3\frac{2k-2}{2k-1}$-approximation algorithm. The first phase of the algorithm is the same as that of Algorithm 2, so it selects $2n-3$ edges. In the second phase of the algorithm at most $n-1$ additional edges are selected, since the selection of the first additional edge already connects two vertices with different colors.

Finally, in the third phase, the algorithm removes some selected edges, thus the found subgraph at the end has indeed at most $2n - 3 + n - 1 = 3n - 4$ edges. Also, if OPT denotes the minimum number of edges of an internally vertex-color-avoiding connected graph then

$$3n - 4 \leq 3n - 3(1 + \frac{1}{2k-1}) = 3n - 3 \cdot \frac{2k-2}{2k-1} \cdot \frac{k}{k-1} =$$

$$= 3 \cdot \frac{2k-2}{2k-1} \cdot \left(\frac{2k-1}{2k-2} \cdot n - \frac{k}{k-1} \right) = 3 \cdot \frac{2k-2}{k-1} \cdot \mathrm{OPT},$$

where the first inequality is true since $k \geq 2$. This means that this is indeed a $3\frac{2k-2}{2k-1}$-approximation algorithm,

Similarly to Algorithm 2, the first and the third phases of this algorithm run in time $O(|C||V|^2)$, and the second phase can clearly be done in time $O(|V|^2)$. Therefore, the running time of the algorithm is $O(|C||V|^2)$. $\square$

We have presented three approximation algorithms, one for each type of color-avoiding-connectivity. The main idea of these algorithms was given by the 2-approximation algorithm presented for the minimal 2-edge connected spanning subgraph problem. It would be interesting to investigate whether other algorithms, for example the algorithm of Cheriyan, Sebo or Vishnu [3, 15, 17], could be modified to provide a better approximation algorithms for these problems.

6 Summary

In this thesis, we provided an overview on the color-avoiding-avoiding connectivity on complex networks, moreover, we proved new theorems about some extremal problems of color-avoiding connected graphs. Namely, we determined the minimum number of edges from which a color-avoiding connected graph can be constructed on a given number of vertices with a given number of colors, and we investigated the problem of determining the maximum number of edges that can be removed such that the graph remains color-avoiding connected. In real-life scenarios, both types of problems naturally occur: in the first one we want to minimize the construction costs, in the second one we want to minimize the maintenance costs.

We proved some interesting results regarding both topics. For the first problem we showed how many edges are needed to construct an edge-, vertex- or internally vertex-color-avoiding connected graph. Furthermore, we also gave constructions, which achieve the lower bounds. For the second problem we proved that deciding whether there exists an edge-, vertex or internally vertex-color-avoiding connected spanning subgraph with a given number of edges is NP-complete. However, we gave polynomial time approximation algorithms for finding an edge-, vertex- or internally vertex-color-avoiding connected spanning subgraph with minimum number of edges.

The thesis contains many new results in the topic of color-avoiding connectivity. These results not just enrich the area, but raise other interesting questions. For example our approximation algorithm is based on a simple algorithm for the corresponding question of the uncolored case. There are a lot more complex and better approximation algorithms, so one might be able to construct a better approximation algorithm for the colored case as well by appropriately modifying an other one. Also it would be interesting to test the algorithms and draw conclusions on important real-life networks like the public transport network of Budapest. Last, but not least one can invert the second type of discussed questions, which leads to a scenario, where there is an existing network which is not color-avoiding connected, and one want to add the least number of edges so that it becomes color-avoiding connected.

References

[1] V. Auletta, Y. Dinitz, Z. Nutov, and D. Parente. A 2-approximation algorithm for finding an optimum 3-vertex-connected spanning subgraph. *Journal of Algorithms*, 32(1):21–30, 1999.

[2] A.-L. Barabási. *Network Science*. Cambridge University Press, 2016.

[3] J. Cheriyan, A. Sebő, and Z. Szigeti. Improving on the 1.5-approximation of a smallest 2-edge connected spanning subgraph. *SIAM Journal on Discrete Mathematics*, 14(2):170–180, 2001.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[5] M. Giusfredi and F. Bagnoli. A self-organized criticality method for the study of color-avoiding percolation. *Internet Science*, 11938, 2019.

[6] M. Giusfredi and F. Bagnoli. From color-avoiding to color-favored percolation in diluted lattices. *Future Internet*, 12(8), 2020.

[7] F. Harary. The maximum connectivity of a graph. *Proceedings of the National Academy of Sciences*, 48(7):1142–1146, 1962.

[8] C. Hunkenschröder, S. Vempala, and A. Vetta. A 4/3-approximation algorithm for the minimum 2-edge connected subgraph problem. *ACM Transactions on Algorithms*, 15(4), 2019.

[9] A. Kadović, S. M. Krause, G. Caldarelli, and V. Zlatić. Bond and site color-avoiding percolation in scale free networks. *Phys. Rev. E*, 2018.

[10] S. Khuller and U. Vishkin. Biconnectivity approximations and graph carvings. *ACM*, 41(2), 1994.

[11] S. M. Krause, M. M. Danziger, and V. Zlatić. Hidden connectivity in networks with vulnerable classes of nodes. *Physical Review X 6*, 2016.

[12] S. M. Krause, M. M. Danziger, and V. Zlatić. Color-avoiding percolation. *Physical Review E 966*, 2017.

[13] I. Kryven. Bond percolation in coloured and multiplex networks. *Nature Communications*, 10, 2019.

[14] R. Molontay and K. Varga. On the complexity of color-avoiding site and bond percolation. In *SOFSEM 2019: Theory and Practice of Computer Science*, volume 11376, 2019.

[15] V. V. Narayan. A 17/12-approximation algorithm for 2-vertex-connected spanning subgraphs on graphs with minimum degree at least 3. 2017. https://arxiv.org/abs/1612.04790.

[16] S. Roy, A. Cherevko, S. Chakraboty, N. Ghosh, and P. Ghosh. Leveraging network science for social distancing to curb pandemic spread. *IEEE Access*, 9:26196–26207, 2021.

[17] J. Sebő, A. Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-tsp, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 43:597–629, 2014.