Parameterized Complexity of Critical Node Cuts

Danny Hermelin[1], Moshe Kaspi[1], Christian Komusiewicz[2], and Barak Navon[1]

1 **Department of Industrial Engineering and Management, Ben-Gurion University, Israel**
hermelin@bgu.ac.il, moshe@exchange.bgu.ac.il, baraknav@post.bgu.ac.il

2 **Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany**
christian.komusiewicz@tu-berlin.de

Abstract

We consider the following graph cut problem called Critical Node Cut (CNC): Given a graph G on n vertices, and two positive integers k and x, determine whether G has a set of k vertices whose removal leaves G with at most x connected pairs of vertices. We analyze this problem in the framework of parameterized complexity. That is, we are interested in whether or not this problem is solvable in $f(\kappa) \cdot n^{O(1)}$ time (*i.e.*, whether or not it is *fixed-parameter tractable*), for various natural parameters κ. We consider four such parameters:

- The size k of the required cut.
- The upper bound x on the number of remaining connected pairs.
- The lower bound y on the number of connected pairs to be removed.
- The treewidth w of G.

We determine whether or not CNC is fixed-parameter tractable for each of these parameters. We determine this also for all possible aggregations of these four parameters, apart from $w + k$. Moreover, we also determine whether or not CNC admits a polynomial kernel for all these parameterizations. That is, whether or not there is an algorithm that reduces each instance of CNC in polynomial time to an equivalent instance of size $\kappa^{O(1)}$, where κ is the given parameter.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases graph cut problem, NP-hard problem, treewidth

Digital Object Identifier 10.4230/LIPIcs.IPEC.2015.343

1 Introduction

In 2013 a polio virus struck Israel. The virus spread in alarming speed, creating a nationwide panic of parents concerned about the well-being of their children. It was obvious to the Israeli health department that vaccinating all Israeli children is not a practical solution in the given time frame. Thus it became clear that some areas of the country should be vaccinated first in order to stop the spread of the virus as quickly as possible. Let us represent a geographic area as a vertex of a graph, and the roads between areas as edges of the graph. In this setting, vaccinating an area corresponds to deleting a certain vertex from the graph. Thus, the objective of stopping the virus from spreading translates to minimizing the number of *connected pairs* (two vertices which are in the same connected component) in the corresponding graph after applying the vaccination.

This scenario can be modeled by the following graph-theoretic problem called Critical Node Cut (CNC). In this problem, we are given an undirected simple graph G and two integers k and x. The objective is to determine whether there exists a set $C \subseteq V(G)$ of

10th International Symposium on Parameterized and Exact Computation (IPEC 2015).
Editors: Thore Husfeldt and Iyad Kanj; pp. 343–354
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

at most k vertices in G, such that the graph $G - C$ which results from removing C from G, contains at most x connected pairs. In this sense, the cut C is considered *critical* since removing it from G leaves few (at most x) connected pairs. For convenience, throughout the paper we will count *ordered* connected pairs; *i.e.*, pairs $(u, v) \in V(G) \times V(G)$, $u \neq v$, where u and v belong to same connected component in $G - C$.

The goal of CNC is thus, roughly speaking, to destroy the connectivity of a given graph as much as possible given a certain budget for deleting vertices. From this point of view, CNC fits nicely to the broad family of *graph-cut problems*. Graph-cut problems have been studied widely and are among the most fundamental problems in algorithmic research. Examples include Min Cut, Max Cut, Multicut, Multiway Cut, Feedback Vertex Set, and Vertex Cover (see *e.g.* [19] for definitions of these problems). The latter is the special case of CNC with $x = 0$. Since Vertex Cover is one of the most important problems in the theory of algorithmic design for NP-hard problems, CNC provides a natural test bed to see which of the techniques from this theory can be extended, and to what extent.

Previous Work and Applications. The CNC problem has been studied from various angles. The problem was shown to be NP-complete in [3] (although its NP-completeness follows directly from the much earlier NP-completeness result for Vertex Cover). In trees, a weighted version of CNC is NP-complete whereas the unweighted version can be solved in polynomial time [12]. The case of bounded treewidth can be solved using dynamic programming in $O(n^{w+1})$ time, where n is the number of vertices in the graph and w is its treewidth [1]. Local search [3] and simulated annealing [28] were proposed as heuristic algorithms for CNC. Finally, in [29] an approximation algorithm based on randomized rounding was developed.

Due to its generic nature, the CNC problem has been considered in various applications. One example application is the virus vaccination problem discussed above [3]. Other interesting applications include protecting a computer/communication network from corrupted nodes, analyzing anti-terrorism networks [23], measuring centrality in brain networks [21], insulin signaling [27], and protein-protein interaction network analysis [6].

Our Results. From reviewing the literature mentioned above, it is noticeable that an analysis of CNC from the perspective of parameterized complexity [13] is lacking. The purpose of this paper is to remedy this situation. We examine CNC with respect to four natural parameters along with all their possible combined aggregations. The four basic parameters we examine are:

- The size k of the solution (*i.e.*, the critical node cut) C.
- The bound x on the number of connected pairs in the resulting graph $G - C$.
- The number of connected pairs y to be removed from G; if G is connected and has n vertices then $y = n(n-1) - x$.
- The treewidth w of G.

Table 1 summarizes all we know regarding the complexity of CNC with respect to these four parameters and their aggregation.

Let us briefly go through some of the trivial results given in the table above. First note that CNC with $x = 0$ is precisely the Vertex Cover problem, which means that CNC is not in FPT (and therefore has no polynomial kernel) for parameter x unless P=NP. This also implies that the problem is unlikely to admit a polynomial kernel even when parameterized by $w+x$, since such a kernel would imply a polynomial kernel for Vertex Cover parameterized by the treewidth w which is known to cause the collapse of the polynomial hierarchy [5, 15].

Table 1 Summary of the complexity results for CRITICAL NODE CUT.

Parameter				Result	
k	x	y	w	FPT	P-Kernel
✓				NO (Thm. 1)	NO (Thm. 1)
	✓			NO	NO
		✓		YES (Thm. 13)	NO (Thm. 14)
			✓	NO (Thm. 5)	NO (Thm. 14)
✓	✓			YES (Thm. 3)	YES (Thm. 4)
✓		✓		YES (Thm. 13)	NO (Thm. 14)
✓			✓	?	NO (Thm. 14)
	✓	✓		YES	YES
	✓		✓	YES (Thm. 12)	NO
		✓	✓	YES (Thm. 13)	NO (Thm. 14)
✓	✓	✓		YES	YES
✓	✓		✓	YES (Thm. 3)	YES (Thm. 4)
✓		✓	✓	YES (Thm. 13)	NO (Thm. 14)
	✓	✓	✓	YES	YES
✓	✓	✓	✓	YES	YES

Next, notice that if our input graph G has no isolated vertices, we have $x + y = \Omega(n)$, and therefore CNC is FPT and has a polynomial kernel for $x + y$ (as isolated vertices can safely be discarded). This of course means that the same applies for parameters $k+x+y$, $x+y+w$, and $k + x + y + w$.

Our first result, stated in Theorem 1, shows that CNC parameterized by k is W[1]-hard. Thus, CNC is unlikely to have an FPT algorithm under this parameterization. We then show in Theorem 3 and Theorem 4, that when considering $x + k$ as a parameter, we can extend two classical VERTEX COVER techniques to the CNC problem. Our main technical result is stated in Theorem 5, where we prove that CNC is W[1]-hard with respect to w, the treewidth of the input graph. This is somewhat surprising since not many graph cut problems are known to be W[1]-hard when parameterized by treewidth. Also, the result complements nicely the $O(n^{w+1})$-time algorithm of [1] by showing that this algorithm cannot be improved substantially. We complement this algorithm from the other direction by showing in Theorem 12 that CNC can be solved in $f(w + x) \cdot n^{O(1)}$ time. Finally, we show in Theorem 13 and Theorem 14 that CNC is FPT with respect to y, and has no polynomial kernel even for $y + w + k$. Due to lack of space, most proofs are deferred to a full version of the article.[1]

Related Work. This paper belongs to a recent extensively explored line of research in parameterized complexity where various types of graph cut problems are analyzed according to various natural problem parameterizations. This line of research can perhaps be traced back to the seminal paper of Marx [24] who studied five such problems, and in the process introduced the fundamental notion of *important separators*. This paper paved the way to several parameterized results for various graph cut problems, including MULTICUT [7, 20, 22, 24, 25, 26, 30], MULTIWAYCUT [9, 10, 11, 20, 24, 30], and STEINER MULTICUT [8]. A particularly closely related problem to CNC is the so-called VERTEX INTEGRITY problem where we want to remove k vertices from a graph such that the largest connected component

[1] A preliminary full version can be obtained at `http://arxiv.org/abs/1503.06321`.

in the remaining graph has a bounded number of vertices. Fellows and Stueckle [18] were the first to analyze this problem from a parameterized point of view; we refer the reader to [14] for a detailed overview. The edge deletion variant of Vertex Integrity has also been studied [16].

2 Parameters k and $k + x$

We now consider the parameters k and $k + x$ for CNC. We first show that the problem is W[1]-hard for k. To this end, we devise a reduction from Clique. From an instance (G, ℓ) of Clique, which asks whether G contains a complete subgraph of order ℓ, we construct H, the graph of our CNC instance, as follows: Replace each edge in G by n parallel edges, and then subdivide each of the new edges once. Next, add an edge in H between each pair of nonadjacent vertices of G. Finally, set $k := \ell$.

▶ **Theorem 1.** Critical Node Cut *is* W[1]*-hard with respect to k.*

We next show that the above result holds also for some restricted subclasses. A *split graph* is a graph in which the vertices can be partitioned into a clique and an independent set. We slightly modify the construction by adding all the edges missing between every pair of non-dummy vertices. In this way, the vertices of G form a clique and the dummy vertices form an independent set, while all arguments in the proof above still hold. For a fixed integer $d \geq 1$, a graph is called *d-degenerate* if each of its subgraphs has a vertex with a degree of at most d. For $d = 1$ (*i.e.*, a forest), the CNC problem has a polynomial algorithm based on dynamic programming [12]. We modify the construction in the proof above by subdividing all the edges except those that are adjacent to dummy vertices. This results in a 2-degenerate graph, and also a bipartite graph with one side containing all vertices of G and the other containing all the dummy vertices. By a slightly more careful (yet still along the same lines) argument it can be shown that the conclusion of Theorem 1 still stands.

▶ **Corollary 2.** Critical Node Cut *remains* W[1]*-hard with respect to k even if the input graph is split, bipartite, or d-degenerate for any fixed $d \geq 2$.*

We next consider the parameter $k + x$. We will show that the basic techniques known for the case of $x = 0$, *i.e.*, Vertex Cover, can be extended to the case where $x > 0$. First, a simple branching strategy can be developed into an FPT algorithm for the parameter $k + x$.

▶ **Theorem 3.** Critical Node Cut *is* FPT *with respect to $k + x$.*

The running time can be improved by using a more elaborate approach in the last step. For example, isolated edges can be dealt with in a dynamic programming subroutine. Then the remaining instance on which brute-force has to be applied has at most $1.5x$ vertices. Next, we show that a simple "high-degree rule" leads to a polynomial kernel.

▶ **Theorem 4.** Critical Node Cut *has a polynomial kernel with respect to $k + x$.*

3 Parameter w

In this section we will show that CNC is unlikely to be fixed-parameter tractable when parameterized by w. This implies that we cannot substantially improve on the $O(n^{w+1})$ algorithm of [1]. Since we will not directly use the notion of treewidth and tree decompositions, we refer to [4] for their definition.

▶ **Theorem 5.** Critical Node Cut *is* W[1]*-hard with respect to the treewidth w of the input graph.*

Our proof of the theorem above is via the well-known multicolored clique technique [17] which utilizes generic gadget structure to construct a reduction from the W[1]-complete Multicolored Clique problem: Given an undirected simple graph G with n vertices and m edges, a coloring function $c : V(G) \to \{1, \ldots, \ell\}$ of the vertices of G, and a parameter ℓ, determine whether G has a clique which includes exactly one vertex from each color. Throughout this section we use (G, c, ℓ) to denote an arbitrary input to Multicolored Clique. As usual in parameterized reductions, we can assume that n and ℓ are sufficiently larger than any fixed constant, and that n is sufficiently larger than ℓ.

In the multicolored clique technique, we construct *selection* gadgets which encode the selection of vertices and edges of G (one per each color class and pair of color classes, respectively), and *validation* gadgets which ensure that the vertices and edges selected indeed form a clique in G. In our reduction below, we will force any feasible solution to delete a large number of vertices from the constructed CNC instance in order to reach the required bound on the number of remaining connected pairs. We will ensure that such a solution always leaves $4\binom{\ell}{2}$ very large components which encode the selection of $\binom{\ell}{2}$ edges in G. The bound on the number of connected pairs will require all these huge components to have equal size, which in turn can only happen if the edges selected in G are edges between the same set of ℓ vertices (implying that these ℓ vertices form a clique in G). In what follows, we use (H, k, x) to denote the instance of CNC that we construct, where H is the input graph, k is the size of the required cut, and x is the bound on the number of connected pairs. Note that for our proof to go through, we will also need to show that the treewidth of H is bounded by some function in ℓ.

Connector gadgets. To each vertex $u \in V(G)$, we assign two unique integer identifiers: $low(u) \in \{1, \ldots, n\}$ and $high(u) \in \{n+1, \ldots, 2n\}$, where $high(u) = 2n + 1 - low(u)$. Our selection gadgets are composed from gadgets which we call *connector gadgets*. A connector gadget *corresponds* to a vertex of G, and can be of *low order* or *high order*. A low order connector gadget corresponding to a vertex $u \in V(G)$ consists of a clique of size ℓ^4 and an independent set of size $n^{16} + low(u)$ which have all edges between them; *i.e.*, it is a complete split graph on these two sets of vertices. Similarly, a high order connector gadget corresponding to $u \in V(G)$ is a complete split graph on a clique of size ℓ^4 and an independent set of size $n^{16} + high(u)$.

We refer to the clique in a connector gadget as the *core* of the gadget, and to the remaining vertices as the *guard* of the gadget. Only vertices in the core will be adjacent to vertices outside the gadget. Notice that the huge independent set in the core contributes to a large number of connected pairs in H, and one can delete all these connected pairs only by adding all core vertices to the solution cut. Below we use this property to help us control solutions for our CNC instance.

Selection gadgets. The graph H consists of a selection gadget for each vertex and edge in G (see Figure 1): For a vertex $u \in V(G)$, we will construct a *u-selection gadget* as follows: First we add a clique U of size ℓ^2 to H, and then we connect all the vertices of U to an additional independent set of n^9 vertices, which we call the *dummy vertices* of the u-selection gadget. We next connect U to $(\ell - 1)$ gadget pairs, one pair for each color $i \in \{1, \ldots, \ell\} \setminus \{c(u)\}$. Each pair consists of a low order and a high order connector gadget corresponding to u. We let $A^i_o[u]$ and $B^i_o[u]$ respectively denote the core and guard of the connector gadget *associated* with color $i \in \{1,\ldots,\ell\} \setminus \{c(u)\}$ and of order $o \in \{low, high\}$. We connect U to each connector gadget by adding all edges between all vertices of U and $A_o^i[u]$, for each $i \in \{1,\ldots,\ell\} \setminus \{c(u)\}$ and $o \in \{low, high\}$.

For an edge $\{u_1,u_2\} \in E(G)$, we will construct a $\{u_1,u_2\}$-*selection gadget* as follows: First we add a vertex which we denote by $\{u_1,u_2\}$ to H. We then connect $\{u_1,u_2\} \in V(H)$ to a low order and a high order connector gadget associated with u_1, and to a low order and a high order connector gadget associated with u_2, by adding all edges between vertex $\{u_1,u_2\} \in V(H)$ and the core vertices of these gadgets. We let $A_o^u[u_1,u_2]$ and $B_o^u[u_1,u_2]$ respectively denote the core and guard of the connector gadget corresponding to $u \in \{u_1,u_2\}$ of order $o \in \{low, high\}$ in the $\{u_1,u_2\}$-selection gadget. Finally, we connect $\{u_1,u_2\} \in V(H)$ to an additional set of n^4 dummy neighbors of degree one in H.

Validation gadgets. We next add the validation gadgets to H, one for each ordered pair of distinct colors (i,j), $i \neq j$. For such a pair (i,j), the (i,j)-validation gadget simply consists of two cliques $V_{low}[i,j]$ and $V_{high}[i,j]$, each of size ℓ^7. The validation is done through the connections of these two cliques to the remainder of the graph. Consider a u-selection gadget for a vertex $u \in V(G)$ of color i. We add all possible edges between $V_{low}[i,j]$ and $A_{low}^j[u]$, and all edges between $V_{high}[i,j]$ and $A_{high}^j[u]$. This is done for every vertex of color i. Consider next a $\{u_1,u_2\}$-selection gadget where $c(u_1) = i$ and $c(u_2) = j$. We add all possible edges between $V_{low}[i,j]$ and $A_{high}^{u_1}[u_1,u_2]$, and all possible edges between $V_{high}[i,j]$ and $A_{low}^{u_1}[u_1,u_2]$. In this way, $V_{low}[i,j]$ is connected to low order connector gadgets of vertex selection gadgets and to high order connector gadgets of edge selection gadgets, and $V_{high}[i,j]$ is connected in the opposite way.

CNC instance. The graph H of our CNC instance is thus composed of $4\binom{\ell}{2}$ validation cliques which have ℓ^7 vertices each, n vertex selection gadgets each of size $(\ell-1)(2n^{16}+2n+1+2\ell^4)+n^9+\ell^2$, and m edge selection gadgets which have $2(2n^{16}+2n+1+2\ell^4)+n^4+1$ vertices each. We finish the description of our reduction by setting k, the size of the required critical node cut, to

$$k := \left(2(\ell-1)n + 4m - 8\binom{\ell}{2}\right) \cdot \ell^4 + \ell^3 + \binom{\ell}{2},$$

and setting x, the bound on the number of connected pairs, to

$$\begin{aligned} x \;\; := \;\; & (n-\ell)\,(n^9+\ell^2)(n^9+\ell^2-1) + \left(m - \binom{\ell}{2}\right)(n^4+1)n^4 + \\ & 4\binom{\ell}{2}(2n^{16}+2n+1+\ell^7+2\ell^4)(2n^{16}+2n+\ell^7+2\ell^4). \end{aligned}$$

► **Lemma 6.** *The graph H has treewidth at most $4\binom{\ell}{2}\ell^7 + \ell^4 + \ell^2$.*

Proof. We use two well known facts about treewidth: The treewidth of a graph is the maximum treewidth of all its components, and adding α vertices to a graph of treewidth at most β results in a graph of treewidth at most $\alpha+\beta$. Using these two facts we get that a connector gadget has treewidth at most ℓ^4, since we add ℓ^4 vertices to a graph of treewidth 0 (the independent set of vertices). From this we conclude that each selection gadget has treewidth at most $\ell^4+\ell^2$, since we either add a clique of size ℓ^2 or a single vertex to a graph whose connected components have treewidth bounded by ℓ^4. Therefore, since H itself is constructed by adding $4\binom{\ell}{2}\cdot\ell^7$ validation vertices to a graph whose connected components have treewidth at most $\ell^4+\ell^2$, the lemma follows. ◄

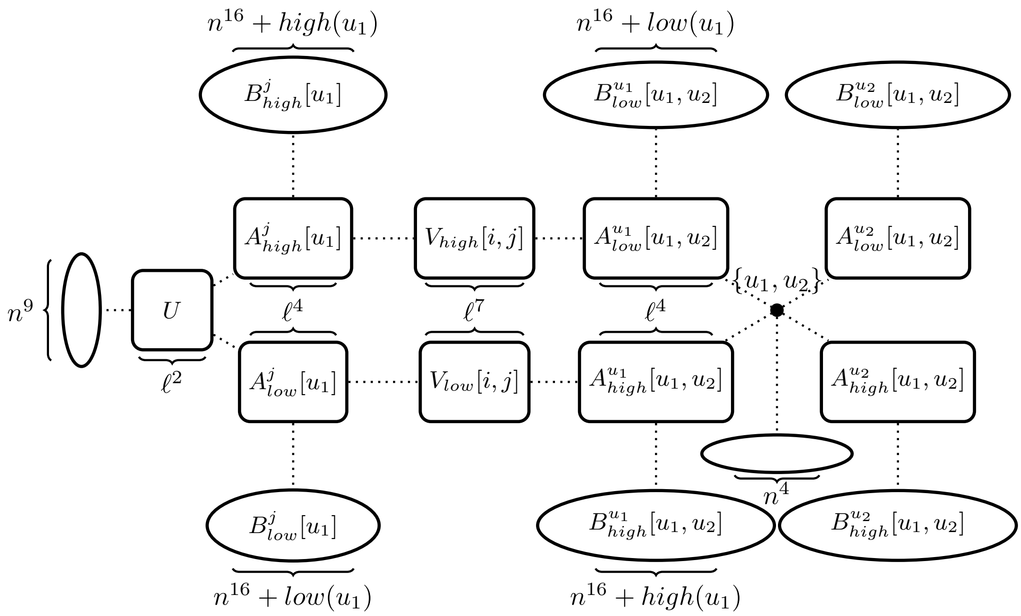

Figure 1 The connection of selection gadgets via a validation gadget. In the example, we consider a vertex $u_1 \in V(H)$ with $c(u_1) = i$ which is adjacent to a vertex $u_2 \in V(H)$ with $c(u_2) = j$. The diagram depicts the pair of low and high connector gadgets associated with color j in the u-selection gadget that are connected to the $\{u_1, u_2\}$-selection gadget. The remaining $(\ell - 2)$ pairs of connector gadgets in the u-selection gadget are not depicted. The rectangle boxes represent cliques and each ellipsis represents an independent set. The dotted lines depict a complete set of edges between two sets of vertices.

From a multicolored clique to a critical node cut. Suppose (G, c, ℓ) has a solution, *i.e.*, a multicolored clique S of size ℓ. Then one can verify that the cut $C \subseteq V(H)$ defined by

$$C := \{U : u \in S\} \cup \{\{u_1, u_2\} : u_1 \neq u_2 \in S\} \cup \left\{v : v \in A^c_o[u], u \notin S\right\} \\ \cup \left\{v : v \in A^u_o[u_1, u_2], u_1 \neq u_2 \notin S\right\}$$

is of size k, and $H - C$ contains exactly three types of non-trivial connected components:
- $n - \ell$ components which include a clique U of size ℓ^2 along with n^9 dummy vertices.
- $m - \binom{\ell}{2}$ components which include a single vertex of $E(G)$ along with n^4 dummy vertices.
- $4\binom{\ell}{2}$ components which have $2n^{16} + 2n + 1 + \ell^7 + 2\ell^4$ vertices each.

Thus, $H - C$ has exactly x connected pairs, and C is indeed a solution to (H, k, x).

From a critical node cut to a multicolored clique. To complete the proof of Theorem 5, we show that if (H, k, x) has a solution, *i.e.*, a cut C of size k where $H - C$ has at most x connected pairs, then G has a multicolored clique of size ℓ. We do this, using a few lemmas that restrict the structure of solutions to our CNC instance. The first one of these, Lemma 7 below, shows that we can restrict our attention to cuts which include only core vertices of connector gadgets and vertices of $V(G) \cup E(G)$.

▶ **Lemma 7.** *If there is a solution to (H, k, x), then there is a solution C to this instance which includes no guard vertices, no dummy vertices, and no validation vertices of H.*

Proof. Let C be a solution to (H, k, x). If C includes any dummy vertex v of H, then since v is a vertex whose neighborhood is a clique, we can either replace v with one of its neighbors (which is a non-dummy vertex) or, if C contains all neighbors of v, we remove v from C. Both modifications of C do not increase the number of connected pairs in $H - C$. Similarly, if C includes guard vertices, these can be safely replaced with core vertices.

Next, we show that C cannot contain any validation clique completely. To this end, note that a core of a connector gadget which is not completely included in C contributes more than n^{32} connected pairs in $H - C$. This can be seen by counting the number of connected pairs between a single core vertex and all of its guard neighbors. Thus, since $(16\binom{\ell}{2}+1)n^{32} > x$ assuming a sufficiently large n, the cut C must include all but at most $16\binom{\ell}{2}$ cores of connector gadgets in H. But as each validation clique is of size $\ell^7 > 8\binom{\ell}{2}\ell^4 + \ell^3 + \binom{\ell}{2}$ (for sufficiently large ℓ), we have $k - \ell^7 < (2(\ell-1)n + 4m - 16\binom{\ell}{2})\ell^4$, which means that if C includes a validation clique it does not include enough cores. Thus, C cannot completely contain any validation clique.

Finally, consider the case that C contains a proper subset of some validation clique $V_o[i, j]$ in H. Observe first that if the validation clique is not completely isolated in $H - C$, then a vertex $v \in C \cap V_o[i, j]$ can be safely replaced by a core vertex that is adjacent to $V_o[i, j]$ as v is not a cut vertex in $H - (C \setminus \{v\})$. Thus, the only remaining case is that all vertices that have a neighbor in $V_o[i, j]$ are in C. Then, deleting the vertices in $V_o[i, j]$ removes at most $\ell^7(\ell^7 - 1)$ connected pairs. By the choice of k, and the number of core vertices, C cannot contain all core vertices. Consider a core vertex $u \notin C$. Since C does not contain any guard vertices, adding u to C removes at least $n^{16} > \ell^7(\ell^7 - 1)$ connected pairs. Thus, we can remove all vertices of $V_o[i, j] \cap C$ from C and replace them by u without increasing the number of connected pairs in $H - C$. Thus, there is a solution that contains no vertices of validation cliques. ◀

Assume that (H, k, x) has a solution, and fix a solution C as in Lemma 7. By the definition of k, we know that the cut C cannot include all connector gadgets. A connector gadget in $H - C$ induces a large number of connected pairs, at least n^{32}, due to the guard vertices of the gadget. Let us therefore call a connected component in $H - C$ *huge* if it contains at least n^{32} connected pairs. The next lemma shows that there can only be a certain number of these huge components in $H - C$, and reveals some further restriction on any solution cut C. We call a maximal non-empty (but not necessarily proper) subset of a core in $H - C$ a *partial core*.

▶ **Lemma 8.** *If C is a solution to (H, k, x) as in Lemma 7, then C includes $(2(\ell-1)n + 4m - 8\binom{\ell}{2})$ cores. Furthermore, there are precisely $4\binom{\ell}{2}$ huge components in $H - C$, each consisting of a validation clique, two partial cores, and the two guard sets of the partial cores.*

Proof. Let $A_1, \ldots, A_t$ denote all partial cores in $H - C$. Note that since each core is of size $\ell^4 > \ell^3 + \binom{\ell}{2}$ (for sufficiently large ℓ), the cut C can include at most $(2(\ell-1)n + 4m - 8\binom{\ell}{2})$ complete cores by definition of k, and so $t \geq 8\binom{\ell}{2}$. By Lemma 7, the graph $H - C$ contains all $4\binom{\ell}{2}$ validation cliques. Let $Q_1, \ldots, Q_s$ denote the components in $H - C$ that contain at least one validation clique, and let $q_i := |Q_i| - 1$ for each i, $1 \leq i \leq s$. Observe that for any huge component Q in $H - C$, we have $Q \in \{Q_1, \ldots, Q_s\}$.

Now, since the total number of validation cliques is $4\binom{\ell}{2}$, we have $s \leq 4\binom{\ell}{2}$, and the total number of connected pairs in all the Q_i's is lower bounded by $\sum_{i=1}^{s} q_i^2$. Note that each partial core A_j belongs to some Q_i and contributes at least $n^{16} + 1$ vertices to its size (accounting for a single vertex of A_j and all its guard neighbors), and therefore at least n^{32} connected pairs. It can now be seen that since $\sum_{i=1}^{s} q_i^2$ is concave and symmetric, it is minimized when the number of addends is as large as possible and all of the addends are of equal size. This happens when $s = 4\binom{\ell}{2}$ and each Q_i includes exactly two A_j's, giving us $\sum_{i=1}^{s} q_i^2 = \sum_{i=1}^{s}((2n^{16})^2) + o(n^{32}) = 16\binom{\ell}{2}n^{32} + o(n^{32})$. If $s < 4\binom{\ell}{2}$ or there is one Q_i that contains more then two A_j's, then the sum will be at least $(16\binom{\ell}{2}+1)n^{32} > x$. It follows that there are exactly $4\binom{\ell}{2}$ huge components, that each have two A_j's. These huge components contribute altogether at least $16\binom{\ell}{2}n^{32}$ connected pairs.

We have thus established that there are $4\binom{\ell}{2}$ huge components in $H - C$, and each includes a validation clique, two partial cores, and the guard sets adjacent to these partial cores which are not in C according to Lemma 7. To see that the huge components contain nothing else, recall first that the overall number of connected pairs in these huge components is at least $16\binom{\ell}{2}n^{32}$. Thus, the number of further additional connected pairs in $H - C$ is at most $x - 16\binom{\ell}{2}n^{32} = n \cdot n^{18} + o(n^{18}) < 2n^{19}$. Now, if A contains other vertices, then by construction it must contain either a vertex from a clique U corresponding to a vertex u of G, or a vertex $\{u, u'\}$ corresponding to an edge of G. In either of these cases, this additional vertex is adjacent to at least n^4 dummy vertices, implying that Q has an additional number of $n^4 \cdot n^{16} = n^{20} > 2n^{19}$ connected pairs, a contradiction. ◀

Slightly smaller than huge components are *large components* in $H - C$ which have at least n^{18} connected pairs and fewer than n^{32} connected pairs. Further smaller are *big components* which have at least n^8 connected pairs, and less than n^{18} connected pairs.

▶ **Lemma 9.** *If C is a solution to (H, k, x) as in Lemma 7, then C includes exactly ℓ cliques $U_1, \ldots, U_\ell$ corresponding to vertices $u_1, \ldots, u_\ell \in V(G)$, and there are precisely $n - \ell$ large components in $H - C$.*

Proof. Note that $x = 16\binom{\ell}{2}n^{32} + (n-\ell)n^{18} + o(n^{18})$. By Lemma 8, we know that $H - C$ contains $4\binom{\ell}{2}$ huge components, and so these already account for $16\binom{\ell}{2}n^{32}$ connected pairs in $H - C$. For every $u \in V(G)$, if the clique corresponding to U is not completely contained in C, then there is a large component corresponding to u in $H - C$, since by Lemma 7, all n^9 dummy neighbors of U are existent in $H - C$. Furthermore, any large component in $H - C$ is of this form. Thus, if C contains $\ell' < \ell$ cliques corresponding to vertices of G, then the number of connected pairs in $H - C$ is at least $16\binom{\ell}{2}n^{32} + (n-\ell')n^{18} > 16\binom{\ell}{2}n^{32} + (n-\ell)n^{18} + o(n^{18}) = x$, a contradiction. Moreover, by our choice of k, the cut C cannot include $(2(\ell-1)n + 4m - 8\binom{\ell}{2})$ cores (as is necessary by Lemma 8) and more than ℓ such cliques U, since $(2(\ell-1)n + 4m - 8\binom{\ell}{2})\ell^4 + (\ell+1)\ell^2 > k$. ◀

▶ **Lemma 10.** *If C is a solution to (H, k, x) as in Lemma 7, then C includes exactly $\binom{\ell}{2}$ vertices which correspond to edges in G, and there are precisely $m - \binom{\ell}{2}$ big components in $H - C$.*

Proof. Let us call each element in the set $\{U \subset V(H) : u \in V(G)\} \cup \{\{u, u'\} \in V(H) : \{u, u'\} \in E(G)\}$ a *G-element*. Thus, each G-element belongs to its unique selection gadget in H, and corresponds to either a vertex or an edge of G. Moreover, each core is adjacent to exactly one G-element. By Lemma 9 we know that C contains ℓ G-elements corresponding to vertices of G. We next argue that it also contains $\binom{\ell}{2}$ G-elements corresponding to edges of G.

Consider a huge component Q in $H - C$. By Lemma 8, Q contains two partial cores A and A' and Q does *not* contain the unique G-element that is adjacent to the two partial cores. Thus, the G-element neighbors of exactly $8\binom{\ell}{2}$ partial cores are contained in C. The set of cliques $U_1, \ldots, U_\ell \subseteq C$ promised by Lemma 9 accounts for at most $2(\ell-1) \cdot \ell = 4\binom{\ell}{2}$ such cores, as each U_i has exactly $2(\ell-1)$ neighboring cores in H. Notice that by the choice of k, after accounting for the vertices in C required by Lemma 8 and Lemma 9, the remaining number of vertices is $\binom{\ell}{2}$. Any G-element representing a vertex has $\ell^2 > \binom{\ell}{2}$ vertices, and thus all remaining deleted G-elements correspond to edges of G. Now observe that each of them can account for at most four partial cores as they have exactly four neighboring cores in H. Consequently, the number of deleted G-elements that correspond to edges in G is at least $\binom{\ell}{2}$. By the choice of k, it is thus exactly $\binom{\ell}{2}$. ◀

▶ **Lemma 11.** *The set of vertices $u_1, \dots, u_\ell$ specified in Lemma 9 induces a multicolored clique in G.*

Proof. Lemma 8, Lemma 9, and Lemma 10 together state that C includes at least $(2(\ell - 1)n + 4m - 8\binom{\ell}{2}) \cdot \ell^4$ core vertices, at least ℓ^3 vertices in cliques corresponding to vertices of G, and at least $\binom{\ell}{2}$ vertices corresponding to edges of G. By our selection of k, all these lower bounds are in fact equalities. Thus, all but ℓ cliques U, $u \in V(G)$, are present in $H - C$, and all but $\binom{\ell}{2}$ edges of G are present in $H - C$. All these vertices contribute at least $(n - \ell)\,(n^9 + \ell^2)(n^9 + \ell^2 - 1) + \left(m - \binom{\ell}{2}\right)(n^4 + 1)n^4$ connected pairs in $H - C$, due to their dummy neighbors. Thus, by definition of x, the total number of connected pairs from huge components in $H - C$ is $4\binom{\ell}{2}(2n^{16} + 2n + 1 + \ell^7 + 2\ell^4)(2n^{16} + 2n + \ell^7 + 2\ell^4)$.

Now, note that according to Lemma 8, $H - C$ includes exactly $8\binom{\ell}{2}$ partial cores with no neighboring G-elements. The set of cliques $U_1, \dots, U_\ell \subseteq C$, promised by Lemma 9, accounts for at most $2(\ell - 1) \cdot \ell = 4\binom{\ell}{2}$ partial cores. Moreover, each clique (corresponding to a vertex) is of a different color, otherwise the specific structure promised by Lemma 8 is violated. Similarly, the $\binom{\ell}{2}$ deleted G-elements that correspond to edges in G, promised by Lemma 10, account for at most $4\binom{\ell}{2}$ partial cores, and each edge corresponds to a different pair of colors. Consequently, the only way to remove the required number of neighboring G-elements is if these upper bounds are met with equality. Thus, we have ℓ vertices and $\binom{\ell}{2}$ edges of different colors, as required in a multicolored clique.

Finally, observe that due to the fact that we have accounted for all the vertices in C, it is clear that each huge component consists of two complete (*i.e.*, non-partial) cores. Thus, the size of each of these huge components is $2n^{16} + \ell^7 + 2\ell^4 + high(u_1) + low(u_1')$ for $u_1, u_1' \in V(G)$. Therefore, the only way for the total number of connected pairs in all huge components to not exceed $4\binom{\ell}{2}(2n^{16} + 2n + 1 + \ell^7 + 2\ell^4)(2n^{16} + 2n + \ell^7 + 2\ell^4)$ is if all huge components have equal size, *i.e.*, exactly $(2n^{16} + 2n + 1 + \ell^7 + 2\ell^4)$ vertices each. But this can happen only if we have $u_1 = u_1'$ in the pair of connector guards $B_o^i[u_1]$ and $B_{\bar{o}}^{u_1'}[u_1', u_2]$, in each huge component of $H - C$, as this is the only way for the guard vertices to sum up to $2n^{16} + 2n + 1$. Consequently, the set of $\binom{\ell}{2}$ edges selected in G are edges between $u_1, \dots, u_\ell$ implying that they indeed form a clique. ◀

4 Parameters $w + x$ and y

If we combine the treewidth parameter w with the parameter for the number of connected pairs x, then we obtain fixed-parameter tractability. This can be derived via an optimization variant of Courcelle's theorem due to Arnborg et al. [2]. Using tree decompositions, we obtain a more efficient algorithm.

▶ **Theorem 12.** *The* Critical Node Cut *problem is* FPT *with respect to $w + x$.*

Finally, we consider the CNC problem parameterized by y. We will show that the problem is FPT under this parameterization but has no polynomial kernel even for the aggregate parameterization of $k + y + w$.

▶ **Theorem 13.** *The* Critical Node Cut *problem is FPT with respect to* y.

▶ **Theorem 14.** *The* Critical Node Cut *problem parameterized by* $k + y + w$ *has no polynomial kernel unless the polynomial hierarchy collapses.*

5 Discussion

We considered a natural graph cut problem called Critical Node Cut (CNC) under the framework of parameterized complexity. The only parameterization left open in our analysis is the parameter $w + k$, and so the first natural question left open in the paper is whether CNC is fixed-parameter tractable under this parameterization (we know it is unlikely that it admits a polynomial kernel). It would also be interesting to see how parameters *maximum degree* and *pathwidth* affect the parameterized complexity of CNC. Finally, one can consider the edge variant of the problem (where one is required to delete edges instead of vertices) and the directed variant of the problem. Several of our proofs do not transfer directly to these variants. For example, it is open whether the edge deletion variant of CNC is W[1]-hard with respect to the number of edge deletions.

Acknowledgments. The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement number 631163.11, and by the ISRAEL SCIENCE FOUNDATION (grant No. 551145/).

References

1 Bernardetta Addis, Marco Di Summa, and Andrea Grosso. Removing critical nodes from a graph: complexity results and polynomial algorithms for the case of bounded treewidth. *Optimization online (www.optimization-online.org)*, 2011.

2 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991.

3 Ashwin Arulselvan, Clayton W Commander, Lily Elefteriadou, and Panos M Pardalos. Detecting critical nodes in sparse graphs. *Computers & Operations Research*, 36(7):2193–2200, 2009.

4 Hans L Bodlaender. A tourist guide through treewidth. *Acta Cybernetica*, 11(1-2):1, 1994.

5 Hans L Bodlaender, Rodney G Downey, Michael R Fellows, and Danny Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009.

6 Vladimir Boginski and Clayton W Commander. Identifying critical nodes in protein–protein interaction networks. *Clustering challenges in biological networks*, pages 153–167, 2009.

7 Nicolas Bousquet, Jean Daligault, and Stéphan Thomassé. Multicut is FPT. In *Proceedings of the 43rd Annual Symposium on Theory of Computing (STOC)*, pages 459–468. ACM, 2011.

8 Karl Bringmann, Danny Hermelin, Matthias Mnich, and Erik Jan van Leeuwen. Parameterized complexity dichotomy for steiner multicut. In *Proceedings of the 32nd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 157–170, 2015.

9 Yixin Cao, Jianer Chen, and Jia-Hao Fan. An $O(1.84^k)$ parameterized algorithm for the multiterminal cut problem. In *Proceedings of the 19th Annual Symposium on Fundamentals of Computation Theory (FCT)*, pages 84–94, 2013.

10 Jianer Chen, Yang Liu, and Songjian Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.

11 Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *ACM Transactions on Computation Theory*, 5(1):3, 2013.

12 Marco Di Summa, Andrea Grosso, and Marco Locatelli. Complexity of the critical node problem over trees. *Computers & Operations Research*, 38(12):1766–1774, 2011.

13 Rodney G Downey and Michael R Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

14 Pål Grønås Drange, Markus Sortland Dregi, and Pim van't Hof. On the computational complexity of vertex integrity and component order connectivity. In *Proceedings of the 25th Annual International Symposium on Algorithms and Computation (ISAAC)*, pages 285–297, 2014.

15 Andrew Drucker. New limits to classical and quantum instance compression. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2012.

16 Jessica Enright and Kitty Meeks. Deleting edges to restrict the size of an epidemic. *CoRR*, abs/1504.05773, 2015.

17 Michael R Fellows, Danny Hermelin, Frances A Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.

18 Michael R Fellows and Sam Stueckle. The immersion order, forbidden subgraphs and the complexity of network integrity. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 6(1):23–32, 1989.

19 Michael R Garey and David S Johnson. Computers and intractability; a guide to the theory of NP-completeness. *San Francisco, LA: Freeman*, 1979.

20 Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011.

21 Karen E Joyce, Paul J Laurienti, Jonathan H Burdette, and Satoru Hayasaka. A new measure of centrality for brain networks. *PLoS One*, 5(8):e12200, 2010.

22 Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Magnus Wahlström. Fixed-parameter tractability of multicut in directed acyclic graphs. *SIAM Journal on Discrete Mathematics*, 29(1):122–144, 2015.

23 Vito Latora and Massimo Marchiori. How the science of complex networks can help developing strategies against terrorism. *Chaos, solitons & fractals*, 20(1):69–75, 2004.

24 Dániel Marx. Parameterized graph separation problems. *Theoretical Computer Science*, 351(3):394–406, 2006.

25 Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transaction on Algorithms*, 9(4):1–30, 2013.

26 Dániel Marx and Igor Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *Proceedings of the 43rd Annual Symposium on Theory of Computing (STOC)*, pages 469–478, 2011.

27 Cullen M Taniguchi, Brice Emanuelli, and C Ronald Kahn. Critical nodes in signalling pathways: insights into insulin action. *Nature Reviews Molecular Cell Biology*, 7(2):85–96, 2006.

28 Mario Ventresca. Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem. *Computers & Operations Research*, 39(11):2763–2775, 2012.

29 Mario Ventresca and Dionne Aleman. A derandomized approximation algorithm for the critical node detection problem. *Computers & Operations Research*, 43:261–270, 2014.

30 Mingyu Xiao. Simple and improved parameterized algorithms for multiterminal cuts. *Theory of Computing Systems*, 46(4):723–736, 2010.