Obtaining a Bipartite Graph by Contracting Few Edges*

Pinar Heggernes[1], Pim van 't Hof[1], Daniel Lokshtanov[2], and Christophe Paul[3]

1   Department of Informatics, University of Bergen, Norway.
     `{pinar.heggernes|pim.vanthof}@ii.uib.no`
2   Department of Computer Science and Engineering,
     University of California San Diego, USA.
     `dlokshtanov@cs.ucsd.edu`
3   CNRS, LIRMM, Université Montpellier 2, France.
     `paul@lirmm.fr`

Abstract

We initiate the study of the BIPARTITE CONTRACTION problem from the perspective of parameterized complexity. In this problem we are given a graph G on n vertices and an integer k, and the task is to determine whether we can obtain a bipartite graph from G by a sequence of at most k edge contractions. Our main result is an $f(k) \, n^{O(1)}$ time algorithm for BIPARTITE CONTRACTION. Despite a strong resemblance between BIPARTITE CONTRACTION and the classical ODD CYCLE TRANSVERSAL (OCT) problem, the methods developed to tackle OCT do not seem to be directly applicable to BIPARTITE CONTRACTION. To obtain our result, we combine several techniques and concepts that are central in parameterized complexity: iterative compression, irrelevant vertex, and important separators. To the best of our knowledge, this is the first time the irrelevant vertex technique and the concept of important separators are applied in unison. Furthermore, our algorithm may serve as a comprehensible example of the usage of the irrelevant vertex technique.

1   Introduction

ODD CYCLE TRANSVERSAL (OCT) is a central problem in parameterized complexity. The establishment of its fixed parameter tractability by Reed, Smith, and Vetta [24] in 2004, settling a long-standing open question [8], supplied the field with the powerful new technique of iterative compression [22]. Both OCT and the closely related EDGE BIPARTIZATION problem take as input a graph G and an integer k, and ask whether a bipartite graph can be obtained by deleting at most k vertices, respectively k edges, from G. These two problems can be viewed as two ways of measuring how close G is to being bipartite. Over the last few years a considerable amount of research has been devoted to studying different measures of how close a graph is to being bipartite [9, 10, 15, 14], and how similarity to a bipartite graph can be exploited [6]. Another natural similarity measure is defined by the

---

BIPARTITE CONTRACTION problem: Given a graph G and an integer k, can we obtain a bipartite graph from G by a sequence of at most k edge contractions in G? Considering the significant amount of interest the problems OCT and EDGE BIPARTIZATION have received, we find it surprising that BIPARTITE CONTRACTION has not yet been studied with respect to parameterized complexity. In this area, a graph problem with input G and k is said to be *fixed parameter tractable (FPT) when parameterized by k* if there is an algorithm with running time $f(k) n^{O(1)}$, where the function f depends only on k and not on the size of G.

The classical computational complexity of contracting at most k edges in a given graph to obtain a graph with a specific structure has been studied by Watanabe et al. [27, 28] and by Asano and Hirata [1]. NP-completeness of BIPARTITE CONTRACTION follows from an easy polynomial-time reduction from EDGE BIPARTIZATION, in which every edge of the input graph is replaced by a path of sufficiently large odd length. The study of edge contractions in general is motivated from Hamiltonian graph theory and graph minor theory, and it has applications in computer graphics and cluster analysis [18]. Graph minors play a central role in parameterized complexity, and the edge contraction operation in turn is essential in the study of graph minors: a graph H is a minor of a graph G if H can be obtained from G by a sequence of edge contractions, edge deletions, and vertex deletions. Although deciding whether a graph H is a minor of a given graph G is FPT when parameterized by the size of H [25], deciding whether H can be obtained from G by edge contractions is NP-complete already for some very small fixed graphs H, such as a path or a cycle on four vertices [2].

In this paper we show that BIPARTITE CONTRACTION is FPT when parameterized by the number k of edges to be contracted. The key ingredients of our algorithm fundamentally differ from the ones used in the above-mentioned algorithms for OCT and EDGE BIPARTIZATION. In the algorithm for OCT by Reed, Smith, and Vetta [24], iterative compression is combined with maximum flow arguments. The recent nearly linear time algorithm for the two problems, due to Kawarabayashi and Reed [14], uses the notion of odd minors, together with deep structural results of Robertson and Seymour [25] about graphs of large treewidth without large clique minors. Interestingly, BIPARTITE CONTRACTION does not seem to be amenable to these approaches.

Although our algorithm is based on iterative compression, it seems difficult to adapt the compression step from [24] for OCT to work for BIPARTITE CONTRACTION. Instead, we perform the compression step using a variant of the *irrelevant vertex technique*, introduced by Robertson and Seymour [25] (see also [26]). In particular, if the treewidth of the input graph is large, then we identify an irrelevant edge that can be deleted from the graph without affecting the outcome. The irrelevant vertex technique has played a key role in the solutions of several problems (see, e.g., [13, 15, 16]).

Our algorithm crucially deviates from previous work in the manner in which it finds the irrelevant edge. While previous work has relied on large minor models as obstructions to small treewidth, ours uses the fact that any graph of high treewidth contains a large *p-connected* set X [7]. A vertex set X is *p-connected* if, for any two subsets X_1 and X_2 of X with $|X_1| = |X_2| \leq p$, there are $|X_1|$ vertex-disjoint paths with one endpoint in X_1 and the other in X_2. Using p-connected sets in order to find irrelevant edges has several advantages. First, our algorithm avoids the huge parameter-dependence which seems to be an inadvertent side effect of applying Robertson and Seymour's graph minors machinery. Second, our arguments are nearly self-contained, and rely only on results whose proofs are simple enough to be taught in a graduate class.

Using p-connected sets in order to find an irrelevant vertex or edge is non-trivial, because p-connectivity is a more "implicit" notion than that of a large minor model. We overcome this

difficulty by using *important sets*. Important sets and the closely related notion of *important separators* were introduced in [20] to prove the fixed parameter tractability of multiway cut problems. The basic idea is that in many problems where terminals need to be separated in some way, it is sufficient to consider separators that are "as far as possible" from one of the terminals. Important separators turned out to be a crucial component, in some cases implicitly, in the solutions of cardinal problems in parameterized complexity [4, 5, 21, 23]. To the best of our knowledge, this is the first time the irrelevant vertex technique and important sets (or separators) are used together. We believe that this combination will turn out to be a useful and powerful tool.

2   Definitions and Notation

All graphs considered in this paper are finite, undirected, and simple, i.e., do not contain multiple edges or loops. Given a graph G, we denote its vertex set by $V(G)$ and its edge set by $E(G)$. We also use the ordered pair $(V(G), E(G))$ to represent G. We let $n = |V(G)|$ and $m = |E(G)|$. For two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the *disjoint union* of G_1 and G_2 is the graph $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$. The *deletion* of an edge $e \in E(G)$ yields the graph $G - e = (V(G), E(G) \setminus e)$. For a set $X \subseteq V(G)$, we write $G[X]$ to denote the subgraph of G *induced* by X. A graph is *connected* if there is a path between each pair of its vertices. The *connected components* of a graph are its maximal connected subgraphs. For any set $X \subseteq V(G)$, we write $\delta_G(X)$ to denote the set of edges in G that have exactly one endpoint in X. We define $d_G(X) = |\delta(X)|$.

The *contraction* of edge xy in G deletes vertices x and y from G, and replaces them by a new vertex, which is made adjacent to precisely those vertices that were adjacent to at least one of the vertices x and y. The resulting graph is denoted G/xy. Every edge contraction reduces the number of vertices in the graph by exactly one. We point out that several edges might disappear as the result of a single edge contraction. For a set $S \subseteq E(G)$, we write G/S to denote the graph obtained from G by repeatedly contracting an edge from S until no such edges remain. Let H be a graph with $V(H) = \{h_1, h_2, \ldots, h_\ell\}$. A graph G is *H-contractible* if H can be obtained from G by contracting edges. Saying that G is H-contractible is equivalent to saying that G has a so-called *H-witness structure* $\mathcal{W}$, which is a partition of $V(G)$ into *witness sets* $W(h_1), W(h_2), \ldots, W(h_\ell)$, satisfying the following properties: each witness set induces a connected subgraph of G, and for every two $h_i, h_j \in V(H)$, there is an edge in G between a vertex of $W(h_i)$ and a vertex of $W(h_j)$ if and only if h_i and h_j are adjacent in H. Let $G' = G[W(h_1)] \cup \cdots \cup G[W(h_\ell)]$ be the graph obtained from G by removing all the edges of G, apart from the ones that have both endpoints in the same witness set. In order to contract G to H, it is necessary and sufficient to contract all the edges of some spanning forest F of G'. Note that $|E(F)| = \sum_{i=1}^{\ell}(|W(h_i)| - 1) = |V(G)| - |V(H)|$.

A *2-coloring* of a graph G is a function $\phi : V(G) \to \{1, 2\}$. We point out that a 2-coloring of G is merely an assignment of colors 1 and 2 to the vertices of G, and should therefore not be confused with a *proper* 2-coloring of G, which is a 2-coloring with the additional property that no two adjacent vertices receive the same color. An edge uv is said to be *good* (with respect to ϕ) if $\phi(u) \neq \phi(v)$, and uv is called *bad* (with respect to ϕ) otherwise. A *good component* of ϕ is the vertex set of a connected component of the graph $(V(G), E')$, where $E' \subseteq E$ is the set of all edges that are good with respect to ϕ. Any 2-coloring ϕ of G defines a partition of $V(G)$ into two sets V_ϕ^1 and V_ϕ^2, which are the sets of vertices of G colored 1 and 2 by ϕ, respectively. A set $X \subseteq V(G)$ is a *monochromatic component* of ϕ if $G[X]$ is a connected component of $G[V_\phi^1]$ or a connected component of $G[V_\phi^2]$. We write

$\mathcal{M}_\phi$ to denote the set of all monochromatic components of ϕ. The *cost* of a 2-coloring ϕ is defined as $\sum_{X \in \mathcal{M}_\phi} (|X| - 1)$. Note that the cost of a 2-coloring ϕ of G is 0 if and only if ϕ is a proper 2-coloring of G.

Let G be a graph. A *tree decomposition* of G is a pair $(T, \mathcal{X} = \{X_t\}_{t \in V(T)})$, where T is a tree and $\mathcal{X}$ is a collection of subsets of $V(G)$, satisfying the following three properties: (1) $\cup_{t \in V(T)} X_t = V$; (2) $\forall uv \in E(G), \exists t \in V(T) : \{u, v\} \subseteq X_t$; and (3) $\forall v \in V(G)$, $T[\{t : v \in X_t\}]$ is connected. The *width* of a tree decomposition is $\max_{t \in V(T)} |X_t| - 1$ and the *treewidth* of G, denoted $tw(G)$, is the minimum width over all tree decompositions of G.

The syntax of *monadic second order (MSO) logic* of graphs includes the logical connectives $\vee, \wedge, \neg$, variables for vertices, edges, sets of vertices and sets of edges, the quantifiers $\forall, \exists$ that can be applied to these variables, and the following five binary relations:

- $u \in U$, where u is a vertex variable and U is a vertex set variable;
- $d \in D$, where d is an edge variable and D is an edge set variable;
- **inc**(d, u), where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident to the vertex u;
- **adj**(u, v), where u and v are vertex variables and the interpretation is that u and v are adjacent;
- equality of variables representing vertices, edges, sets of vertices and sets of edges.

3   Bipartite Contraction and the Cost of 2-Colorings

In the BIPARTITE CONTRACTION problem we are given a graph G and an integer k, and the task is to determine whether there exists a set $S \subseteq E(G)$ of at most k edges such that G/S is bipartite. The following lemma allows us to reformulate this problem in terms of 2-colorings of the graph G.

▶ **Lemma 1.** *A graph G has a 2-coloring ϕ of cost at most k if and only if there exists a set $S \subseteq E(G)$ of at most k edges such that G/S is bipartite.*

Proof. Suppose G has a 2-coloring ϕ of cost at most k. We build an edge set S as follows. For each monochromatic component $X \in \mathcal{M}_\phi$, find a spanning tree T_X of $G[X]$ and add the $|X| - 1$ edges of T_X to S. The total number of edges in S is exactly the cost of ϕ, so $|S| \leq k$. It remains to argue that $G' = G/S$ is bipartite. Note that G' is obtained from G by contracting each spanning tree T_X to a single vertex t_X. Let ϕ' be the 2-coloring of G' that assigns to each $t_X \in V(G')$ the color of the corresponding monochromatic component $X \in \mathcal{M}_\phi$. Since each monochromatic component has been contracted to a single vertex, G' has no edge that is bad with respect to ϕ'. Hence ϕ' is a proper 2-coloring of G', implying that G' is bipartite.

For the reverse direction, suppose there is a set $S \subseteq E(G)$ of at most k edges such that $G' = G/S$ is bipartite. We define G^* to be the graph with the same vertex set as G and edge set S, i.e., $G^* = (V(G), S)$. Let $\mathcal{W}$ be the G'-witness structure of G whose witness sets are exactly the connected components of G^*. Let ϕ' be a proper 2-coloring of G'. We construct a 2-coloring ϕ of G as follows. For every $v \in V(G)$, we set $\phi(v) = \phi'(y)$, where y is the vertex in $V(G')$ such that $v \in W(y)$. Since the monochromatic components of ϕ are exactly the connected components of the graph G^*, and since G^* contains exactly $|S|$ edges, the cost of ϕ is at most $|S| \leq k$.   ◀

An instance of the CHEAP COLORING problem consists of a graph G and an integer k, and the task is to decide whether G has a 2-coloring of cost at most k. Lemma 1 shows that the problems BIPARTITE CONTRACTION and CHEAP COLORING are equivalent.

The deletion of an edge can not increase the cost of a 2-coloring, and can only decrease the cost of a 2-coloring by at most one. We state this as the following observation.

▶ **Observation 1.** Let ϕ be a 2-coloring of G of cost k. For any edge $uv \in E(G)$, the cost of ϕ in $G - uv$ is k or $k - 1$.

Observation 1 allows us to use the well-known *iterative compression* technique of Reed, Smith and Vetta [24] to reduce the CHEAP COLORING problem to the CHEAPER COLORING problem. The CHEAPER COLORING problem takes as input a graph G, an integer k, and a 2-coloring ϕ of G of cost $k + 1$, and the task is to either find a 2-coloring of G of cost at most k, or to conclude that such a coloring does not exist.

▶ **Lemma 2.** *If there is an algorithm for* CHEAPER COLORING *that runs in time* $f(k) n^c$ *for some constant c, then there is an algorithm for* CHEAP COLORING *that runs in time* $f(k) n^c m$.

Proof. Suppose there exists an algorithm for CHEAPER COLORING that runs in time $f(k) n^c$. Then we can solve an instance (G, k) of CHEAP COLORING by iterating over the edges $e_1, e_2, \ldots e_m$ of G as follows. For every $i \in \{1, \ldots, m\}$, we define G_i to be the graph with vertex set $V(G)$ and edge set $E_i = \{e_j \ : \ j \leq i\}$. The graph G_1 has a 2-coloring ϕ_1 of cost 0, which is at most k. For the first k iterations, we trivially maintain a 2-coloring of cost at most k. Now, in iteration i of the algorithm, assume that we have a 2-coloring ϕ_i of cost at most k in G_i. By Observation 1, the cost of ϕ_i in G_{i+1} is at most $k + 1$. If the cost of ϕ_i in G_{i+1} is at most k, then we proceed to iteration $i + 1$. Otherwise, we run the algorithm for CHEAPER COLORING with input (G_{i+1}, k, ϕ_i). If the algorithm concludes that G_{i+1} has no 2-coloring of cost at most k, then, by Observation 1, neither does G. If, on the other hand, the algorithm outputs a 2-coloring ϕ_{i+1} of G_{i+1} of cost at most k, then we proceed to the $(i + 1)$th iteration. Since we call the algorithm for CHEAPER COLORING at most m times, each time with parameter k, the time bound follows.   ◀

We have now almost reached the variant of the problem that will be the focus of attention in the remainder of this paper. Given a graph G and two disjoint vertex sets $T_1, T_2 \subseteq V(G)$, a 2-coloring ϕ of G is a (T_1, T_2)-*extension* if ϕ colors every vertex in T_1 with 1 and every vertex in T_2 with 2. In the CHEAP COLORING EXTENSION problem we are given a *bipartite* graph G, two integers k and t, and two disjoint vertex sets T_1 and T_2 such that $|T_1| + |T_2| \leq t$; the sets T_1 and T_2 are not related to the sets of the bipartition of G. The objective is to find a (T_1, T_2)-extension ϕ of cost at most k, or to conclude that such a 2-coloring does not exist. We will say that a (T_1, T_2)-extension ϕ is a *cheapest* (T_1, T_2)-extension if there is no (T_1, T_2)-extension ϕ' with strictly lower cost than ϕ.

▶ **Lemma 3.** *If there is an algorithm for* CHEAP COLORING EXTENSION *that runs in time* $f(k, t) n^c$ *for some constant c, then there is an algorithm for* CHEAPER COLORING *that runs in time* $4^{k+1} f(k, 2k + 2) n^c$.

Proof. Given an $f(k, t) n^c$ time algorithm for CHEAP COLORING EXTENSION, we show how to solve an instance (G, k, ϕ) of CHEAPER COLORING. Let S be the set of all bad edges in G with respect to ϕ, and let X be the set of endpoints of the edges in S. Since ϕ has cost $k + 1$, we have $|X| \leq 2k + 2$. We create 4^{k+1} instances of CHEAP COLORING EXTENSION as follows.

For every possible partition of X into two sets X_1 and X_2, we set $k' = k$ and $t = |X|$, and we build a graph $G(X_1, X_2)$ from G in the following way. As long as there is an edge $uv \in S$ such that u and v are both in X_1 or both in X_2, contract the edge uv, put the new vertex resulting from the contraction into the set X_i that u and v belonged to, and decrease

k' by 1. Since the cost of ϕ is at most $k+1$, we contract at most $k+1$ edges in this way, and hence $k' \geq -1$. When there are no such edges left, then we discard this partition of X into X_1 and X_2 if $k' = -1$; otherwise, we continue to build an instance of CHEAP COLORING EXTENSION as follows. Delete all edges $uv \in S$ with $u \in X_i$ and $v \in X_j$ such that $i \neq j$. Since S contains all the edges of G that are bad with respect to ϕ, and each of the edges of S is either contracted or deleted, the resulting graph $G(X_1, X_2)$ has no bad edges with respect to ϕ and is therefore bipartite. Thus we obtain an instance $(G(X_1, X_2), k', t, X_1, X_2)$ of CHEAP COLORING EXTENSION with $k' \geq 0$.

We now show that (G, k, ϕ) is a yes-instance of CHEAPER COLORING if and only if there is a partition of X into X_1 and X_2 such that $(G(X_1, X_2), k', t, X_1, X_2)$ with $k' \geq 0$ is a yes-instance of CHEAP COLORING EXTENSION.

Suppose that (G, k, ϕ) is a yes-instance of CHEAPER COLORING. Then there exists a 2-coloring ϕ^* of G of cost at most k. Let X_1 and X_2 be the vertices of X that are colored 1 and 2 by ϕ^*, respectively. Consider the set $S' \subseteq S$ of edges that were contracted in order to obtain $G(X_1, X_2)$ from G in the way described earlier. Since every edge in S' is bad with respect to ϕ^*, the cost of ϕ^* decreased by 1 with every edge contraction. Hence, ϕ^* is an (X_1, X_2)-extension of $G(X_1, X_2)$ of cost k'. We conclude that $(G(X_1, X_2), k', t, X_1, X_2)$ is a yes-instance of CHEAP COLORING EXTENSION.

For the reverse direction, suppose there is a partition of X into X_1 and X_2 such that $(G(X_1, X_2), k', t, X_1, X_2)$ is a yes-instance of CHEAP COLORING EXTENSION with $k' \geq 0$, i.e., the bipartite graph $G(X_1, X_2)$ has an (X_1, X_2)-extension ψ of cost at most k'. Let $S' \subseteq S$ be the set of edges that were contracted in G to create the instance $(G(X_1, X_2), k', t, X_1, X_2)$. Since $k' = k - |S'| \geq 0$, we have that $|S'| \leq k$. We define a 2-coloring θ of G by coloring both endpoints of every edge uv in S' with the color that ψ assigned to the vertex resulting from the contraction of the edge uv, and coloring all other vertices in G with the color they received from ψ. Clearly, the cost of θ is at most $k' + |S'| = k$, and therefore (G, k, ϕ) is a yes-instance of CHEAPER COLORING.

Since we need to run the $f(k, t) \, n^c$ time algorithm for CHEAP COLORING EXTENSION at most 4^{k+1} times, with parameters $k' \leq k$ and $t = |X| \leq 2k + 2$ at each iteration, the time bound follows.   ◄

The next section is devoted to showing that CHEAP COLORING EXTENSION is fixed parameter tractable when parameterized by k and t. The reason we want to work with the CHEAP COLORING EXTENSION problem rather than with the BIPARTITE CONTRACTION problem directly is that, as we shall see in Section 4.2, CHEAP COLORING EXTENSION is a "cut" problem, and is therefore amenable to techniques based on *important separators* [20].

4   Solving Cheap Coloring Extension in FPT Time

In this section, we present an algorithm for the CHEAP COLORING EXTENSION problem. For the remainder of this section, let (G, k, t, T_1, T_2) be a given instance of CHEAP COLORING EXTENSION, where G is assumed to be connected. Recall that G is bipartite. The high level structure of our algorithm is as follows. If the treewidth of G is bounded by a function of k and t, then we can use standard dynamic programming techniques to solve the problem in time $f(k, t) \, n$. If, on the other hand, the treewidth of G is large, then we can find a large set of vertices which is "highly connected". In this case we show how to find in $f(k, t) \, n^{O(1)}$ time an edge $e \in E(G)$, such that G has a (T_1, T_2)-extension of cost at most k if and only if $G - e$ does. We then re-run our algorithm on $G - e$.

To make the distinction between the two cases in our algorithm more precise, we use the

following notion, due to Diestel et al. [7]. A set $X \subseteq V(G)$ is *p-connected* in G if $|X| \geq p$ and, for all subsets $X_1, X_2 \subseteq X$ with $|X_1| = |X_2| \leq p$, there are $|X_1|$ vertex-disjoint paths in G with one endpoint in X_1 and the other in X_2. Diestel et al. [7] prove the following statement in the proof of Proposition 3 (ii): if $h \geq p$ and G contains no p-connected set of size h, then G has treewidth $< h + p - 1$. (In fact, they prove a stronger version of this statement using the notion of an *externally p*-connected set, but we do not need this stronger assertion for our purposes.) We define a set X to be *well-connected* if it is $|X|/2$-connected. Using this definition, the result of Diestel et al. [7] can be seen to imply the following.

▶ **Theorem 4** ([7]). *If $tw(G) > w$, then G contains a well-connected set of size at least $2w/3$.*

The proof of Theorem 4 is constructive. In fact, given G and w, a tree decomposition of width at most w or a well-connected set of size at least $2w/3$ can be computed in time $c^w n^{O(1)}$ for some constant c [7]. We use Theorem 4 to compute either a tree-decomposition of G of width at most $3(4k^2) \, t \, 4^{4k^2} + 3$ or a well-connected set Y of size at least $2(4k^2) \, t \, 4^{4k^2} + 2$. Section 4.1 deals with the first case, whereas the second case is covered in Section 4.2.

4.1 Small Treewidth

Suppose our algorithm has found a tree-decomposition of G of width at most $3(4k^2) \, t \, 4^{4k^2} + 3$. We use the following celebrated theorem by Courcelle [3] to solve the CHEAP COLORING EXTENSION problem in this case.

▶ **Theorem 5** ([3]). *There is an algorithm that tests whether a monadic second order formula ψ holds on a graph G of treewidth w, in time $f(|\psi|, w) \, n$.*

Since CHEAP COLORING EXTENSION can be expressed in monadic second order logic (we omit the details due to page restrictions), we have the following result.

▶ **Lemma 6.** *There is an algorithm that, given an instance (G, k, t, T_1, T_2) of CHEAP COLORING EXTENSION together with a tree-decomposition of G of width w, solves the instance in time $f(k, t, w) \, n$.*

We would like to remark that, given an instance (G, k, t, T_1, T_2) of CHEAP COLORING EXTENSION together with a tree-decomposition of G of width w, it is possible to solve that instance in time $(w + 1)^{O(w)} n$ using standard dynamic programming techniques, which gives a much faster algorithm than the one obtained by applying Theorem 5 on the monadic second order formula.

4.2 Large Treewidth and Irrelevant Edges

Suppose our algorithm did not find a tree-decomposition of G of small width, but instead found a well-connected set Y of size at least $2(4k^2) \, t \, 4^{4k^2} + 2$. We use Y throughout this section to refer to this specific set. An edge $e \in E(G)$ is said to be *irrelevant* if it satisfies the following property: G has a (T_1, T_2)-extension of cost at most k if and only if $G - e$ does. We will show that the presence of the large well-connected set Y guarantees the presence of an irrelevant edge e in G. Hence we find such an irrelevant edge e in G, delete it from the graph, and solve CHEAP COLORING EXTENSION on the instance $(G - e, k, t, T_1, T_2)$. Since each iteration of this process deletes an edge, we will find a tree-decomposition of the graph under consideration of small width after at most m iterations, in which case we solve the problem as described in Section 4.1.

▶ **Observation 2.** Let ϕ be a 2-coloring of G. No bad edge has both endpoints in the same good component of ϕ.

Proof. Suppose, for contradiction, that G has a bad edge uv such that both u and v belong to a good component C of ϕ. Since uv is bad, we have $\phi(u) = \phi(v)$. Every good component is connected, so there is a path P in C, starting in u and ending in v, consisting only of good edges. The path P must contain an even number of edges, implying that P and uv together form an odd cycle in G. This contradicts the assumption that G, which is part of the instance (G, k, t, T_1, T_2) of CHEAP COLORING EXTENSION that we are solving, is bipartite. ◀

▶ **Observation 3.** Let $uv \in E(G)$. If ϕ is a cheapest (T_1, T_2)-extension of $G - uv$ and u and v are in the same good component of ϕ, then uv is irrelevant.

Proof. Suppose u and v belong to the same good component of a cheapest (T_1, T_2)-extension ϕ of $G - uv$. Note that ϕ is a 2-coloring of G, and that the edge uv in G is good with respect to ϕ as a result of Observation 2. Hence ϕ is a (T_1, T_2)-extension of G, and the cost of ϕ in G equals the cost of ϕ in $G - uv$. As a result of Observation 1, ϕ must be a cheapest (T_1, T_2)-extension of G. Since the cost of a cheapest (T_1, T_2)-extension of $G - uv$ equals the cost of a cheapest (T_1, T_2)-extension of G, the edge uv is irrelevant by definition. ◀

In order to use Observation 3, we need to identify two adjacent vertices u and v in G that will end up in the same good component of some cheapest (T_1, T_2)-extension of $G - uv$. The vertices in Y are good candidates, because they are so highly connected to each other. Over the next few lemmas we formalize this intuition. We start with two observations that will allow us, in the proof of Lemma 7 below, to bound the number of bad edges and the number of good components of a cheapest (T_1, T_2)-extension of G of cost at most k.

▶ **Observation 4.** Let ϕ be a 2-coloring of G. If ϕ has cost at most k, then there are less than $2k^2$ bad edges.

Proof. Let $\mathcal{M}'_\phi = \{X \in \mathcal{M}_\phi \ : \ |X| \geq 2\}$ be the set of monochromatic components of ϕ containing more than one vertex, and let G' be the disjoint union of the graphs induced in G by the elements of $\mathcal{M}'_\phi$, i.e., $G' = \bigcup_{X \in \mathcal{M}'_\phi} G[X]$. By definition, the cost of ϕ is $\sum_{X \in \mathcal{M}_\phi}(|X| - 1) = \sum_{X \in \mathcal{M}'_\phi}(|X| - 1)$, which is exactly the number of edges in any spanning forest of G'. Since any forest on at most k edges without isolated vertices has at most $2k$ vertices, we have $|V(G')| \leq 2k$. Every bad edge has both endpoints in $V(G')$, so the number of bad edges is at most $\binom{2k}{2} = 2k^2 - k < 2k^2$. ◀

▶ **Observation 5.** Let ϕ be a cheapest (T_1, T_2)-extension of G. Every good component of ϕ contains a vertex from $T_1 \cup T_2$.

Proof. Suppose a good component C of ϕ does not contain any vertex from $T_1 \cup T_2$. We build a coloring ϕ' from ϕ by changing the color of every vertex in C, leaving the color of every other vertex unchanged, i.e., $\phi'(v) = 3 - \phi(v)$ if $v \in C$, and $\phi'(v) = \phi(v)$ if $v \notin C$. Since $\phi(v) = \phi'(v)$ for every $v \in T_1 \cup T_2$, ϕ' is a (T_1, T_2)-extension of G. Furthermore, every edge that was good with respect to ϕ is good with respect to ϕ', while every edge in $\delta_G(C)$ was bad with respect to ϕ and is good with respect to ϕ'. Recall that G is assumed to be connected. Hence there is some vertex $v \in C$ which is incident to at least one edge that was bad with respect to ϕ. On the other hand, all edges incident to v are good with respect to ϕ'. Hence $\{v\}$ is a monochromatic component of ϕ', but $\{v\}$ was not a monochromatic component of ϕ. This means that $|\mathcal{M}_\phi| < |\mathcal{M}_{\phi'}|$. This, together with the observation that the number of edges that are bad with respect to ϕ' is not higher than the number of edges

that were bad with respect to ϕ, implies that the cost of ϕ' is strictly less than the cost of ϕ. This contradicts the assumption that ϕ is a cheapest (T_1, T_2)-extension of G.   ◄

The next lemma shows that almost all the vertices of Y appear in the same good component of any cheapest (T_1, T_2)-extension ϕ of G of cost at most k. In fact, we prove that the same holds if we remove any edge of G.

▶ **Lemma 7.** *Let $uv \in E(G)$ and let ϕ be a cheapest (T_1, T_2)-extension of $G - uv$. If ϕ has cost at most k, then there exists exactly one good component C^* of ϕ satisfying $|Y \backslash C^*| \leq 2k^2$, and every other good component C' of ϕ satisfies $|Y \cap C'| \leq 2k^2$.*

Proof. Suppose ϕ has cost at most k, and let C be a good component of ϕ. We first show that either $|Y \backslash C| \leq 2k^2$ or $|Y \cap C| \leq 2k^2$. Suppose for contradiction that $|Y \cap C| > 2k^2$ and $|Y \backslash C| > 2k^2$. We define Y_1 to be the smallest of the two sets $Y \cap C$ and $Y \backslash C$, and Y_2 to be any subset of the largest of the two sets such that $|Y_2| = |Y_1|$. Note that $2k^2 + 1 \leq |Y_1| = |Y_2| \leq |Y|/2$. By the definition of a well-connected set, there are $|Y_1| \geq 2k^2 + 1$ vertex-disjoint paths with one endpoint in $Y \cap C$ and the other in $Y \backslash C$. At least $2k^2$ of these paths exist in $G - uv$, and each of those must contain an edge in $\delta_{G-uv}(C)$. Since each edge in $\delta_{G-uv}(C)$ is bad, it follows that ϕ has at least $2k^2$ bad edges, contradicting Observation 4.

Now suppose for contradiction that ϕ does not have a good component C^* with $|Y \backslash C^*| \leq 2k^2$. Then $|Y \cap C| \leq 2k^2$ for every good component C of ϕ, as we showed earlier. Since ϕ has at most $t = |T_1| + |T_2|$ good components as a result of Observation 5, at most $t\,2k^2$ vertices of Y appear in good components. The fact that the size of Y is much larger than $t\,2k^2$, together with the observation that every vertex of G appears in a good component by definition, yields the desired contradiction. Hence we know that ϕ has a good component C^* with $|Y \backslash C^*| \leq 2k^2$. The uniqueness of C^* follows from the sizes of Y and C^*, and the fact that the good components of ϕ are pairwise disjoint.   ◄

There are two problems with how to exploit the knowledge obtained from Lemma 7. The first is that, even though we know that almost all the vertices of Y appear in the same good component together, we do not know exactly which ones do. The second problem is that we are looking for an *edge* with both endpoints in the same good component, and Y could be an independent set and thus not immediately give us an edge to delete. We deal with both problems by employing the very useful notion of *important sets*. For two vertices $x, y \in V(G)$, we say that a set $X \subseteq V(G)$ is (x, y)-*important* if it satisfies the following three properties: (1) $x \in X$ and $y \notin X$; (2) $G[X]$ is connected; and (3) there is no $X' \supset X$, $y \notin X'$ such that $d_G(X') \leq d_G(X)$ and $G[X']$ is connected. The following theorem was first proved in [4]. We use here the formulation in [19], because that one best fits the purposes of this paper.

▶ **Theorem 8** ([4, 19]). *Let x, y be two vertices in a graph G. For every $p \geq 0$, there are at most 4^p (x, y)-important sets X such that $d_G(X) \leq p$. Furthermore, these important sets can be enumerated in time $4^p \cdot n^{O(1)}$.*

Suppose $G - uv$ has a cheapest (T_1, T_2)-extension ϕ of cost at most k for an edge $uv \in E(G)$. We will use the important sets together with Lemma 7 to identify vertices in Y which must be in the unique good component C^* of ϕ that contains all but at most $2k^2$ vertices of Y. We first build a graph G^* from G by adding a new vertex y^* and making y^* adjacent to all vertices in Y. We then enumerate all $x \in T_1 \cup T_2$ and all (x, y^*)-important sets X in G^* such that $d_{G^*}(X) \leq 4k^2$. By Theorem 8, this can be done in time $4^{4k^2} n^{O(1)}$ for each choice

of x. Finally, we define the set Z to be the union of all enumerated sets X. In other words,

$$Z = \{w \in V(G^*) : \exists x \in T_1 \cup T_2, X \subseteq V(G^*), w \in X, d_{G^*}(X) \leq 4k^2, X \text{ is } (x, y^*)\text{-important}\}$$

Observe that, given G and Y, Z can be computed in time $t\, 4^{4k^2} n^{O(1)}$. We will use the set Z in the following way. First we show that if there is an edge $uv \in E(G)$ such that neither u nor v belongs to Z, then the edge uv is irrelevant. Then we show that such an edge always exists.

▶ **Lemma 9.** *Let $uv \in E(G)$ such that $u \notin Z$ and $v \notin Z$. Then uv is irrelevant.*

Proof. Let ϕ be a cheapest (T_1, T_2)-extension of $G - uv$, and suppose ϕ has cost at most k. Let C^* be a good component of ϕ such that $|Y \setminus C^*| \leq 2k^2$. By Lemma 7, such a component C^* exists, and every other good component C of ϕ satisfies $|Y \cap C| \leq 2k^2$. We prove that both u and v are in C^*. Suppose $u \notin C^*$. Then $u \in C$ for some other good component of ϕ, since by definition every vertex belongs to some good component, possibly of size 1. Now C induces a connected subgraph in $G - uv$, and all edges leaving C in $G - uv$ are bad with respect to ϕ by the definition of a good component. Since ϕ has less than $2k^2$ bad edges by Observation 4, it follows that $d_{G-uv}(C) < 2k^2$, and thus $d_G(C) \leq 2k^2$. Furthermore, because $|Y \cap C| \leq 2k^2$, we have that $d_{G^*}(C) \leq 4k^2$. Finally, by Observation 5, C must contain a vertex $x \in T_1 \cup T_2$. Hence there must be an (x, y^*)-important set X such that $C \subseteq X$ and $d(X) \leq 4k^2$ in G^*. But $C \subseteq X \subseteq Z$, which implies that $u \in Z$, contradicting the assumption that $u \notin Z$. The proof that $v \in C^*$ is identical. We conclude that both u and v belong to the good component C^*, and hence, by Observation 3, uv is irrelevant. ◀

▶ **Lemma 10.** *G contains an edge uv such that $u \notin Z$ and $v \notin Z$.*

Proof. We first prove that $d_G(Z) \leq (4k^2)\, t\, 4^{4k^2}$ and $|Z \cap Y| \leq (4k^2)\, t\, 4^{4k^2}$. For the first inequality, it suffices to show that $d_{G^*}(Z) \leq (4k^2)\, t\, 4^{4k^2}$, because G is a subgraph of G^*. By Theorem 8, there exist at most 4^{4k^2} (x, y^*)-important sets with $d_{G^*}(X) \leq 4k^2$ for each $x \in T_1 \cup T_2$. Since Z is the union of all those sets over all elements of $T_1 \cup T_2$, and $|T_1 \cup T_2| \leq t$, the first inequality follows. To see that $|Z \cap Y| \leq (4k^2)\, t\, 4^{4k^2}$, observe that each (x, y^*)-important set X in G^* with $d_{G^*}(X) \leq 4k^2$ contains at most $4k^2$ vertices of Y, since each vertex in Y is a neighbour of y^*.

In order to prove that G contains an edge uv with $u \notin Z$ and $v \notin Z$, we arbitrarily choose two disjoint subsets Y_1, Y_2 of Y such that $Z \cap Y \subseteq Y_2$ and $|Y_1| = |Y_2| = (4k^2)\, t\, 4^{4k^2} + 1$. Since $|Y| \geq 2(4k^2)\, t\, 4^{4k^2} + 2$ by assumption and we showed that $|Z \cap Y| \leq (4k^2)\, t\, 4^{4k^2}$, such sets Y_1, Y_2 always exist. By the definition of a well-connected set, there are $|Y_1| = (4k^2)\, t\, 4^{4k^2} + 1$ vertex-disjoint paths starting in Y_1 and ending Y_2. For every $1 \leq i \leq |Y_1|$, let $u_i v_i$ be the first edge on the ith such path, with $u_i \in Y_1$. Recall that $Z \cap Y \subseteq Y_2$ by assumption. Since all of the u_i's are in Y_1, none of them are in Z. Thus, each v_i that belongs to Z contributes one to $d_G(Z)$, as then $u_i v_i \in \delta_G(Z)$. Since we bounded $d_G(Z)$ from above by $(4k^2)\, t\, 4^{4k^2}$ at the start of this proof, not every v_i can belong to Z. Hence there is an edge $u_i v_i$ with neither endpoint in Z. ◀

We are now ready to state the main lemma of this section.

▶ **Lemma 11.** CHEAP COLORING EXTENSION *can be solved in time $f(k, t)\, n^{O(1)}$.*

Proof. Let (G, k, t, T_1, T_2) be an instance of CHEAP COLORING EXTENSION, and let f be an appropriate function that does not depend on n. We first apply Theorem 4 and the remark immediately following it to compute, in time $f(k, t)\, n^{O(1)}$, either a tree-decomposition of G of width at most $3(4k^2)\, t\, 4^{4k^2} + 3$ or a well-connected set Y of size at least $2(4k^2)\, t\, 4^{4k^2} + 2$.

If we get a tree-decomposition of small width, we apply Lemma 6 to solve the problem in additional time $f(k, t)\, n$. If we find a well-connected set Y, we continue to find an irrelevant edge $e \in E(G)$ and delete it from G. Lemmas 9 and 10 guarantee that such an edge always exists. In order to find e, we first compute the set Z. We already argued that this can be done in time $f(k, t)\, n^{O(1)}$ if G and Y are given. We can find an irrelevant edge as explained in the proof of Lemma 10 in additional polynomial time, since this amounts to computing $Z \cap Y$, choosing Y_2 to contain the whole intersection and as many more vertices as needed to obtain $|Y_2| = (4k^2)\, t\, 4^{4k^2} + 1$, choosing Y_1 to be any subset of $Y \setminus Y_2$ such that $|Y_1| = |Y_2|$, and checking all edges leaving Y_1 to find one whose endpoints do not belong to Z. The total running time of this whole procedure is clearly $f(k, t)\, n^{O(1)}$.

After an irrelevant edge e is deleted from G, we run the whole procedure on $(G - e, k, t, T_1, T_2)$. This can be repeated at most $|E(G)| = n^{O(1)}$ times, and hence the total running time $f(k, t)\, n^{O(1)}$ follows. ◄

Our main result immediately follows from Lemmas 1, 2, 3, and 11.

▶ **Theorem 12.** BIPARTITE CONTRACTION *is fixed parameter tractable when parameterized by* k.

We end this section with a remark on the running time. If we use Lemma 6 in the proof of Lemma 11, then the parameter dependence of the whole algorithm is dominated by a very large function in k [3]. However, as we remarked after Lemma 6, we can obtain a running time of $(4^{O(k^2)})^{4^{O(k^2)}} n = 2^{2^{O(k^2)}} n$ for the small treewidth case in the proof of Lemma 11. This is because the treewidth of the instance at hand is $4^{4k^2} k^{O(1)} = 4^{4k^2 + O(\log k)} = 4^{O(k^2)}$ when the small treewidth case applies. This gives a total running time of of $2^{2^{O(k^2)}} n^{O(1)}$ for our algorithm for BIPARTITE CONTRACTION.

5   Concluding Remarks

We showed that BIPARTITE CONTRACTION is fixed parameter tractable. A highly relevant question is whether this problem admits a *polynomial kernel*, i.e., a polynomial time algorithm that transforms an instance (G, k) into an equivalent instance (G', k') of size $g(k)$, where g is a polynomial in k. Very recently, Kratsch and Wahlström [17] announced that the problems OCT and EDGE BIPARTIZATION admit *randomized* polynomial kernels, which can be obtained using a novel kernelization approach based on matroid theory.

We conclude with the following question: Can some of the algorithms that currently use Robertson-Seymour machinery to find an irrelevant vertex, be modified in such a way that they find an irrelevant vertex using p-connected sets instead?

─── **References** ───────────────────────

1   T. Asano and T. Hirata. Edge-contraction problems. *Journal of Computer and System Sciences*, 26:197–208, 1983.
2   A. E. Brouwer and H. J. Veldman. Contractibility and NP-completeness. *Journal of Graph Theory*, 11:71–79, 1987.
3   B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.
4   J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.
5   J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *Journal of the ACM*, 55(5), 2008.

6   E. Demaine, M. Hajiaghayi and K. Kawarabayashi. Decomposition, approximation, and coloring of odd-minor-free graphs. In *Proceedings of SODA 2010*, 329–344, ACM-SIAM.

7   R. Diestel, K. Yu. Gorbunov, T. R. Jensen and C. Thomassen. Highly connected sets and the excluded grid theorem. *Journal of Combinatorial Theory, Series B*, 75:61–73, 1999.

8   R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science, Springer-Verlag, 1999.

9   J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. *Journal of Computer and System Sciences*, 72:1386–1396, 2006.

10   F. Hüffner. Algorithm engineering for optimal graph bipartization. *Journal of Graph Algorithms and Applications*, 13(2):77–98, 2009.

11   R.M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, 85–103, Plenum Press, New York, 1972.

12   K. Kawarabayashi. Planarity allowing few error vertices in linear time. In *Proceedings of FOCS 2009*, 639–648, IEEE.

13   K. Kawarabayashi. An improved algorithm for finding cycles through elements. In *Proceedings of IPCO 2008*, LNCS 5035:374–384, Springer.

14   K. Kawarabayashi and B. A. Reed. An (almost) linear time algorithm for odd cycles transversal. In *Proceedings of SODA 2010*, 365–378, ACM-SIAM.

15   K. Kawarabayashi and B. A. Reed. Odd cycle packing. In *Proceedings of STOC 2010*, 695–704, ACM.

16   K. Kawarabayashi and P. Wollan. A simpler algorithm and shorter proof for the graph minor decomposition. In *Proceedings of STOC 2010*, 687–694, ACM.

17   S. Kratsch and M. Wahlström. Compression via matroids: a randomized polynomial kernel for Odd Cycle Transversal. Manuscript available as *CoRR abs/1107.3068*.

18   A. Levin, D. Paulusma, and G. J. Woeginger. The computational complexity of graph contractions I: polynomially solvable and NP-complete cases. *Networks*, 51:178–189, 2008.

19   D. Lokshtanov and D. Marx Clustering with local restrictions. In *Proceedings of ICALP 2011*, LNCS 6755:785–797, Springer.

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

21   D. Marx and I. Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *Proceedings of STOC 2011*, 469–478, ACM.

22   R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.

23   I. Razgon and B. O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *Journal of Computer and System Sciences*, 75(8):435–450, 2009.

24   B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32:299–301, 2004.

25   N. Robertson and P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63:65–110, 1995.

26   N. Robertson and P. D. Seymour. Graph Minors. XXII. Irrelevant vertices in linkage problems. Submitted.

27   T. Watanabe, T. Ae, and A. Nakamura. On the removal of forbidden graphs by edge-deletion or edge-contraction. *Discrete Applied Mathematics*, 3:151–153, 1981.

28   T. Watanabe, T. Ae, and A. Nakamura. On the NP-hardness of edge-deletion and edge-contraction problems. *Discrete Applied Mathematics*, 6:63–78, 1983.

29   S. Wernicke. *On the algorithmic tractability of single nucleotide polymorphism (SNP) analysis and related problems.* Diplomarbeit, WSI für Informatik, Universität Tübingen, 2003.