Contracting bipartite graphs to paths and cycles

Konrad K. Dabrowski, Daniël Paulusma*

School of Engineering and Computing Sciences, Durham University, Science Laboratories, South Road, Durham DH1 3LE, United Kingdom

ARTICLE INFO

Article history:
Received 12 June 2017
Accepted 29 June 2017
Available online 5 July 2017
Communicated by Marcin Pilipczuk

Keywords:
Edge contraction
Bipartite graph
Path
Computational complexity

ABSTRACT

Testing if a given graph $G$ contains the $k$-vertex path $P_k$ as a minor or as an induced minor is trivial for every fixed integer $k \geq 1$. However, the situation changes for the problem of checking if a graph can be modified into $P_k$ by using only edge contractions. In this case the problem is known to be NP-complete even if $k = 4$. This led to an intensive investigation for testing contractibility on restricted graph classes. We focus on bipartite graphs. Heggernes, van ’t Hof, Lévêque and Paul proved that the problem stays NP-complete for bipartite graphs if $k = 6$. We strengthen their result from $k = 6$ to $k = 5$. We also show that the problem of contracting a bipartite graph to the 6-vertex cycle $C_6$ is NP-complete. The cyclicity of a graph is the length of the longest cycle the graph can be contracted to. As a consequence of our second result, determining the cyclicity of a bipartite graph is NP-hard.

© 2017 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Algorithmic problems for deciding whether the structure of a graph $G$ appears as a “pattern” within the structure of another graph $H$ are well studied. Here, the definition of a pattern depends on the set $S$ of graph operations that we are allowed to use. Basic graph operations include vertex deletion $d_v$, edge deletion $d_e$ and edge contraction $e_c$. Contracting an edge $uv$ means that we delete the vertices $u$ and $v$ and introduce a new vertex with neighbourhood $(N(u) \cup N(v)) \setminus \{u, v\}$ (note that no multiple edges or self-loops are created in this way). A graph $G$ contains a graph $H$ as a minor if $H$ can be obtained from $G$ using operations from $S = \{d_v, d_e, e_c\}$. For $S = \{d_v, e_c\}$ we say that $G$ contains $H$ as an induced minor, and for $S = \{e_c\}$ we say that $G$ contains $H$ as a contraction.

For a fixed graph $H$ (that is, $H$ is not part of the input), the corresponding three decision problems are denoted by $H$-MINOR, $H$-INDUCED MINOR and $H$-CONTRACTIBILITY, respectively.

A celebrated result by Robertson and Seymour [16] states that the $H$-Minor problem can be solved in cubic time for every fixed pattern graph $H$. The problems $H$-INDUCED MINOR and $H$-CONTRACTIBILITY are harder. Fellows et al. [7] gave an example of a graph $H$ on 68 vertices for which $H$-INDUCED MINOR is NP-complete, whereas Brouwer and Veldman [4] proved that $H$-CONTRACTIBILITY is NP-complete even when $H = P_k$ or $H = C_4$ (the graphs $C_k$ and $P_k$ denote the cycle and path on $k$ vertices, respectively). Both complexity classifications are still not settled, as there are many graphs $H$ for which the complexity is unknown (see also [13]).

We observe that $P_k$-INDUCED MINOR and $C_k$-INDUCED MINOR are polynomial-time solvable for all $k$; it suffices to check if $G$ contains $P_k$ as an induced subgraph, that is, if $G$ is not $P_k$-free, or if $G$ contains an induced cycle of length at least $k$. In order to obtain similar results to those for minors and induced minors, we need to restrict the input of
the $P_k$-CONTRACTIBILITY and $C_k$-CONTRACTIBILITY problems to some special graph class.

Of particular relevance is the closely related problem of determining the cyclicity \cite{10} of a graph, that is, the length of a longest cycle to which a given graph can be contracted. Cyclicity was introduced by Blum \cite{3} under the name co-circularity, due to a close relationship with a concept in topology called circularity (see also \cite{1}). Later Hammack \cite{10} coined the current name for the concept and gave both structural results and polynomial-time algorithms for a number of special graph classes. He also proved that the problem of determining the cyclicity is NP-hard for general graphs \cite{11}.

Van 't Hof, Paulusma and Woeginger \cite{14} proved that the $P_6$-CONTRACTIBILITY problem is NP-complete for $P_6$-free graphs, but polynomial-time solvable for $P_5$-free graphs. Their results can be extended in a straightforward way to obtain a complexity dichotomy for $P_k$-CONTRACTIBILITY restricted to $P_k$-free graphs except for one missing case, namely when $k = 5$ and $\ell = 6$. Fiala, Kamiński and Paulusma \cite{6} proved that $P_4$-CONTRACTIBILITY is NP-complete on line graphs (and thus for claw-free graphs) for $k \geq 7$ and polynomial-time solvable on claw-free graphs (and thus for line graphs) for $k \leq 4$. The problems of determining the computational complexity for the missing cases $k = 5$ and $k = 6$ were left open. The same authors also proved that $C_k$-CONTRACTIBILITY is NP-complete for claw-free graphs \cite{11}, which implies that determining the cyclicity of a claw-free graph is NP-hard.

Hammack \cite{10} proved that $C_k$-CONTRACTIBILITY is polynomial-time solvable on planar graphs for every $k \geq 3$. Later, Kamiński, Paulusma and Thilikos \cite{15} proved that $H$-CONTRACTIBILITY is polynomial-time solvable on planar graphs for every triangle-free graph $H$. Golovach, Kratsch and Paulusma \cite{9} proved that the $H$-CONTRACTIBILITY problem is polynomial-time solvable on AT-free graphs for every triangle-free graph $H$. Hence, as $C_3$-CONTRACTIBILITY is readily seen to be polynomial-time solvable for general graphs, $C_k$-CONTRACTIBILITY and $P_k$-CONTRACTIBILITY are polynomial-time solvable on AT-free graphs for every integer $k \geq 3$. Heggernes et al. \cite{12} proved that $P_k$-CONTRACTIBILITY is polynomial-time solvable on chordal graphs for every $k \geq 1$. Later, Belmonte et al. \cite{2} proved that $H$-CONTRACTIBILITY is polynomial-time solvable on chordal graphs for every graph $H$. Heggernes et al. \cite{12} also proved that $P_6$-CONTRACTIBILITY is NP-complete even for the class of bipartite graphs.

1.1. Research question

We consider the class of bipartite graphs, for which we still have a limited understanding of the $H$-CONTRACTIBILITY problem. In contrast to a number of other graph classes, as discussed above, bipartite graphs are not closed under edge contraction, which means that getting a handle on the $H$-CONTRACTIBILITY problem is more difficult. We therefore focus on the $H = P_k$ and $H = C_k$ cases of the following underlying research question for $H$-CONTRACTIBILITY restricted to bipartite graphs:

Do the computational complexities of $H$-CONTRACTIBILITY for general graphs and bipartite graphs coincide for every graph $H$?

This question belongs to a more general framework, where we aim to research whether for graph classes not closed under edge contraction, one is still able to obtain “tractable” graphs $H$, for which the $H$-CONTRACTIBILITY problem is NP-complete in general. For instance, claw-free graphs are not closed under edge contraction. However, there does exist a graph $H$, namely $H = P_4$, such that $H$-CONTRACTIBILITY is polynomial-time solvable on claw-free graphs and NP-complete for general graphs. Hence, being claw-free at the start is a sufficiently strong property for $P_4$-CONTRACTIBILITY to be polynomial-time solvable, even though applying contractions might take us out of the class. It is not known whether being bipartite at the start is also sufficiently strong.

1.2. Our contribution

We recall that the $H$-CONTRACTIBILITY problem is already NP-hard if $H = C_4$ or $H = P_4$. Hence, with respect to our research question we will need to consider small graphs $H$. While we do not manage to give a conclusive answer, we do improve upon the aforementioned result from Heggernes et al. \cite{12} on bipartite graphs by showing in Section 3 that even $P_5$-CONTRACTIBILITY is NP-complete for bipartite graphs.

**Theorem 1.** $P_5$-CONTRACTIBILITY is NP-complete for bipartite graphs.

We also have the following result, which we prove in Section 4.

**Theorem 2.** The $C_k$-CONTRACTIBILITY problem is NP-complete for bipartite graphs.

We observe that if a graph can be contracted to $C_k$ for some integer $k \geq 3$, it can also be contracted to $C_{\ell}$ for any integer $3 \leq \ell \leq k$. Hence, as an immediate consequence of Theorem 2, we obtain the following result.

**Corollary 1.** The problem of determining whether the cyclicity of a bipartite graph is at least 6 is NP-complete.

2. A known lemma

A graph $G$ contains a graph $H$ as a contraction if and only if for every vertex $h$ in $V_H$ there is a nonempty subset $W(h) \subseteq V_G$ of vertices in $G$ such that:

- $G[W(h)]$ is connected;
- the set $V = \{W(h) \mid h \in V_H\}$ is a partition of $V_G$; and
- for every $h, i \in V_H$, if $h, i \in V_H$, there is at least one edge between the witness sets $W(h_i)$ and $W(h_j)$ in $G$ if and only if $h_i$ and $h_j$ are adjacent in $H$.

The set $W(h)$ is an $H$-witness set of $G$ for $h$, and $W$ is said to be an $H$-witness structure of $G$. If for every $h \in V_H$ we contract the vertices in $W(h)$ to a single vertex, then we obtain the graph $H$. Witness sets $W(h)$ may not be uniquely defined, as there could be different sequences of
edge contractions that modify $G$ into $H$. A pair of vertices $(u,v)$ of a graph $G$ is $P_2$-suitable for some integer $\ell \geq 3$ if and only if $G$ has a $P_2$-witness structure $W$ with $W(p_1) = \{u\}$ and $W(p_\ell) = \{v\}$, where $P_\ell = p_1 \ldots p_\ell$. See Fig. 1 for an example.

**Lemma 1 ([14])**. For $\ell \geq 3$, a graph $G$ is $P_\ell$-contractible if and only if $G$ has a $P_\ell$-suitable pair.

### 3. The proof of Theorem 1

In this section we prove that $P_5$-Contractibility is NP-complete for bipartite graphs. The $P_5$-Contractibility problem restricted to bipartite graphs is readily seen to be in NP. Hence what remains is to prove NP-hardness.

Let $(Q,S)$ be a hypergraph, where $Q$ is some set of elements and $S$ is a set of hyperedges, which are subsets of $Q$. A 2-colouring of $(Q,S)$ is a partition $(Q_1, Q_2)$ of $Q$ with $Q_1 \cap S \neq \emptyset$ and $Q_2 \cap S \neq \emptyset$ for every $S \in S$. The corresponding decision problem is called Hypergraph 2-Colourability and is well known to be NP-complete (see [8]). Just as in the proof of [4] for NP-hardness of $P_4$-Contractibility for general graphs, we will reduce from Hypergraph 2-Colourability. In fact, just as the construction in the proof [12] for $P_6$-Contractibility for bipartite graphs, our construction borrows elements from [4], but is more advanced.

Let $(Q,S)$ be a hypergraph with $Q = \{q_1, \ldots, q_m\}$ and $S = \{S_1, \ldots, S_n\}$. We may assume without loss of generality that $n \geq 2$, $S_i \neq \emptyset$ for each $S_i$ and $S_n = Q$. Given the pair $(Q,S)$, we will construct a graph $G = (V,E)$ in the following way; see Fig. 2 for an example.

- Construct the incidence graph of $(Q,S)$. This is a bipartite graph with partition classes $Q$ and $S$, and an edge between two vertices $q_i$ and $S_j$ if and only if $q_i \in S_j$.
- Add a set $S' = \{S'_1, \ldots, S'_n\}$ of $n$ new vertices. Add an edge between $q_i$ and $S'_j$ if and only if $q_i \in S_j$. We say that $S'_j$ is a copy of $S_j$ and say that it represents a hyperedge that contains the same elements as $S_j$.
- Add an edge between every $S_j$ and $S'_j$, that is, the subgraph induced by $S \cup S'$ is complete bipartite.
- Subdivide each edge $q_i S_j$, that is, remove the edge $q_i S_j$ and replace it by a new vertex $q'_i$ with edges $q'_i q_i$ and $q'_i S_j$. Let $Q'$ consist of all the vertices $q'_i$.
- Add three new vertices $q^*, u_1$ and $u_2$ and edges $q^* u_1$, $q^* u_2$.
- Add an edge between $q^*$ and every $q'_i$.
- Add an edge between $u_1$ and every $S_j$, and an edge between $u_2$ and every $S_j$.
- Add two new vertices $v$ and $w$. Add the edges $u_1 v$ and $u_2 v$, and also an edge between $w$ and every $S'_j$.

The distance between two vertices in a graph is the number of edges of a shortest path between them. The diameter of a graph is the maximum distance over all pairs of vertices in it. We note that the graph $G$ may have arbitrarily large induced paths (alternating between vertices in $Q$ and $S$). However, as we will check in the proof of Lemma 3, $G$ has diameter 4, and this property will be crucial. We first prove the following lemma.

**Lemma 2.** The graph $G$ is bipartite.
Table 1
The (maximum) distances between two different (types of) vertices in G.

|   | u₁ | u₂ | v | w | S | S' | Q | Q' | q' *
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>u₁</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>u₂</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>v</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>w</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S'</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Proof. We partition V into A = {q*, v, w} ∪ S ∪ Q, and B = {u₁, u₂} ∪ S' ∪ Q', and note that G contains no edge between any two vertices in A and no edge between any two vertices in B.

Lemma 3. The hypergraph (Q, S) has a 2-colouring if and only if the graph G contains P₃ as a contraction.

Proof. Let P be on a path on five vertices p₁, . . . , p₅ in that order. First suppose that (Q, S) has a 2-colouring (Q₁, Q₂). We define W(p₁) = {v}, W(p₂) = {q*, u₁, u₂}, W(p₃) = S ∪ Q₁ ∪ Q', W(p₄) = S' ∪ Q₂ and W(p₅) = {w}. We note that the sets W(p₁), . . . , W(p₅) are pairwise disjoint. Moreover, not only W(p₁), W(p₂) and W(p₅), but also W(p₃) and W(p₄) induce connected subgraphs of G, as Sₙ and Sₙ' are connected to every vertex in Q by definition (either via a path of length 2 containing a vertex of Q or directly via an edge). We also observe that there are no edges between vertices from W(p₁) and vertices from W(p₃) ∪ W(p₄) ∪ W(p₅), no edges between vertices from W(p₂) and vertices from W(p₄) ∪ W(p₅) and no edges between vertices from W(p₃) and vertices from W(p₅). We combine these observations with the existence of edges (for instance, u₁v₁, u₁S₁, S₁Sₙ and S₂w) between the two consecutive sets W(pᵢ) and W(pᵢ+1) for i = 1, . . . , 4 to conclude that the sets W(p₁), . . . , W(p₅) form a P₃-witness structure of G.

Now suppose that G contains P₃ as a contraction. Then, by Lemma 1, we find that G has a P₃-witness structure W', where W(p₁) = {x} and W(p₅) = {y} for some vertices x and y. We refer to Table 1 for the distances between vertices of different types. In this table, entries for a vertex and a set or for two sets display the maximum possible distance between them. For instance, the entry for S and Q is 2, as the maximum distance between a vertex in S and a vertex in Q is 2. We also note, for instance, that the distance between any two vertices in Q is 2, because Sₙ' is adjacent to every vertex of Q.

From Table 1 we can see that there are three possible choices for the pair (x, y), which must be of distance at least 4 from each other in G, namely {x, y} = {v, q₁} for any q₁, {x, y} = {q*, w} or {x, y} = {v, w}. We discuss each of these cases below.

Case 1. {x, y} = {v, q₁}.
Let x = v and y = q₁. From Table 1 we find that {u₁, u₂} ≤ W(p₂). Moreover, S ≤ W(p₃), as every vertex in S is of distance 2 from both v and q₁. As W(p₂) induces a connected subgraph by definition and u₁ is not adjacent to u₂, this means that q₁ must be in W(p₂). However, this is not possible as q* is of distance 2 from q₁, which is in W(p₅). Hence Case 1 is not possible.

Case 2. {x, y} = {q*, w}.
Let x = q* and y = w. From Table 1 we find that Q' ∪ {u₁, u₂} ≤ W(p₂) and that S' ≤ W(p₄). The latter, combined with the fact that every vertex of S is adjacent to every vertex of S', implies that S ∩ W(p₂) = ∅. Any path from a vertex in Q' to a vertex in {u₁, u₂} must contain at least one vertex of S ∪ {q*}. As (S ∪ {q*}) ∩ W(p₂) = ∅, this means that W(p₂) does not induce a connected subgraph. This violates the definition of a witness structure, so Case 2 is not possible either.

Case 3. {x, y} = {v, w}.
Let x = v and y = w. From Table 1 we find that {u₁, u₂} ≤ W(p₂). Moreover, S ≤ W(p₃), as every vertex in S is of distance 2 from both v and w. As W(p₂) must induce a connected subgraph of G by definition, this means that q* ∈ W(p₂). From Table 1 we also find that S' ≤ W(p₄).

By definition, W(p₃) must induce a connected subgraph. Recall that S is an independent set. Hence, for each Sᵢ, we find that W(p₃) contains at least one vertex not in S that connects Sᵢ to the other vertices of S (recall that by assumption, n ≥ 2, so there is at least one other vertex in S not equal to Sᵢ). As {u₁, u₂} ≤ W(p₂) and S' ≤ W(p₄), such a vertex can only be in Q and we denote it by q(Sᵢ). As every vertex in Q' has only three neighbours and one of them is q*, which is in W(p₂), we find that q(Sᵢ) must be adjacent to a vertex q(Sⱼ) ∈ Q ∩ W(p₂) in order to connect Sᵢ to the other vertices of S. Note that q(Sᵢ) = q(Sⱼ) is possible for two vertices Sᵢ and Sⱼ with k ≠ j.

The set W(p₃) also induces a connected subgraph and S' is an independent set of size at least 2. Hence, for each Sⱼ, we find that W(p₄) contains at least one vertex not in S' that connects Sⱼ to the other vertices of S'. As S ≤ W(p₃) and w ∈ W(p₃), such a vertex can only be in Q, and we denote it by q(Sⱼ). Note that q(Sⱼ') = q(Sⱼ) is possible for two vertices Sⱼ and Sⱼ' with k ≠ j.

Let Q₁ be the subset of Q that contains all vertices q(Sⱼ), so Q₁ is contained in W(p₃). Similarly, let Q₂ be the subset of Q that contains all vertices q(Sⱼ'), so Q₂ is contained in W(p₄). Each hyperedge Sⱼ contains q(Sⱼ) due to the edges Sⱼq(Sⱼ) and q(Sⱼ)q(Sⱼ'). Moreover, each hyperedge Sⱼ contains q(Sⱼ) due to the edge Sⱼ'q(Sⱼ) and because Sⱼ' is a copy of Sⱼ. Hence Sⱼ contains both an element from Q₁ and an element from Q₂. Moreover, Q₁ and Q₂ are disjoint. Hence, (Q₁, Q₂) is a 2-colouring of (Q, S) (note that there may be elements of Q not in Q₁ ∪ Q₂; we can add such elements to either Q₁ or Q₂ in an arbitrary way). This completes the proof of Lemma 3.

Combining Lemmas 2 and 3 with the aforementioned observation on membership in NP implies Theorem 1.

Theorem 1 (repeated). P₃-Contractibility is NP-complete for bipartite graphs.
4. The proof of Theorem 2

In this section we prove Theorem 2. We do this as follows. Consider the graph G constructed in Section 3 for a given instance (Q, S) of HYPERGRAPH 2-COLOURING. We remove the vertices q* and u2, and instead add a new vertex x that we make adjacent to both v and w. This yields the graph G' = (V', E').

Removing the edge vx from G' results in the graph G' − vx, which is used in the hardness construction of Hegernes et al. [12] for proving that P6-CONTRACTIBILITY is NP-complete.

Lemma 4 ([12]). The hypergraph (Q, S) has a 2-colouring if and only if G' − vx contains P6 as a contraction.

We continue by proving two lemmas for G' that are similar to the two lemmas of Section 3.

Lemma 5. The graph G' is bipartite.

Proof. We partition V' into A' = {v, w} ∪ S ∪ Q, and B' = {u1, x} ∪ S' ∪ Q', and note that G' contains no edge between any two vertices in A' and no edge between any two vertices in B'. □

Lemma 6. The hypergraph (Q, S) has a 2-colouring if and only if the graph G' contains C6 as a contraction.

Proof. Let C be a cycle on six vertices c1, . . . , c6 in that order. First suppose that (Q, S) has a 2-colouring (Q1, Q2). We define the following witness sets: W(c1) = {v}, W(c2) = {u1}, W(c3) = S ∪ Q1 ∪ Q', W(c4) = S' ∪ Q2, W(c5) = {w} and W(c6) = {x}. The sets W(c1), . . . , W(c6) are readily seen to form a C6-witness structure of G'.

Now suppose that G' contains C6 as a contraction. The only vertex of distance at least 3 from S'' in G' is v (in particular recall that S'' is adjacent to every vertex of Q). Hence we may assume without loss of generality that W(c1) = {v} and S'' ∈ W(c4). Then, as the only two neighbours of v are u1 and x, we may also assume without loss of generality that u1 ∈ W(c2) and x ∈ W(c6). Since v and w are the only two neighbours of x, and w is a neighbour of S'' ∈ W(c4), this means that w ∈ W(c5) and thus W(c6) = {x}. The fact that W(c1) = {v} and W(c6) = {x} implies that G' − vx contains P6 as a contraction and we may apply Lemma 4. □

Combining Lemmas 5 and 6 with the observation that C6-CONTRACTIBILITY belongs to NP implies Theorem 2.

Theorem 2 (restated). The C6-CONTRACTIBILITY problem is NP-complete for bipartite graphs.

5. Future work

We have proved that the P5-CONTRACTIBILITY problem is NP-complete for the class of bipartite graphs, which strengthens a result in [12], where NP-completeness was shown for P6-CONTRACTIBILITY restricted to bipartite graphs. As P3-CONTRACTIBILITY is readily seen to be polynomial-time solvable for general graphs, this leaves us with one stubborn open case, namely P4-CONTRACTIBILITY.

Open Problem 1. Determine the complexity of P4-CONTRACTIBILITY for bipartite graphs.

One approach for settling Open Problem 1 would be to first consider chordal bipartite graphs, which are bipartite graphs in which every induced cycle has length 4. We believe that this is an interesting question on its own.

Open Problem 2. Determine the complexity of P4-CONTRACTIBILITY for chordal bipartite graphs.

We also proved that the C6-CONTRACTIBILITY problem is NP-complete for bipartite graphs, which implied that determining the cyclicity of a bipartite graph is NP-hard. As mentioned, C3-CONTRACTIBILITY is polynomial-time solvable for general graphs. This leaves us with the following two open cases.

Open Problem 3. Determine the complexity of Ck-CONTRACTIBILITY for bipartite graphs when 4 ≤ k ≤ 5.

The 2-DISJOINT CONNECTED SUBGRAPHS problem takes as input a graph G and two disjoint subsets Z1 and Z2 of V(G). It asks whether V(G) can be partitioned into sets A1 and A2, such that Z1 ⊆ A1, Z2 ⊆ A2 and both A1 and A2 induce connected subgraphs of G. Telle and Villanger [17] gave an O*(1.7804k)-time algorithm for solving this problem, which is known to be NP-complete even if |Z1| = 2 [14]. Here, the O* notation suppresses factors of polynomial order.

By using the algorithm of [17] as a subroutine and Lemma 1 we immediately obtain an O*(1.7804k)-time algorithm for solving P4-CONTRACTIBILITY on general n-vertex graphs. That is, we guess two non-adjacent vertices u and v with non-intersecting neighbourhoods N(u) and N(v) to be candidates for a P4-suitable pair and then solve the 2-DISJOINT CONNECTED SUBGRAPHS problem for the graph G − {u, v} with Z1 = N(u) and Z2 = N(v) (note that we need to consider at most k2 choices of pairs u, v).

Proposition 1. P4-CONTRACTIBILITY can be solved in O*(1.7804k) time for (general) graphs on n vertices.

The proof of the aforementioned NP-completeness result for 2-DISJOINT CONNECTED SUBGRAPHS in [14] can be modified to hold for bipartite graphs by subdividing each edge in the hardness construction. This brings us to our final open problem.

Open Problem 4. Does there exist an exact algorithm for P4-CONTRACTIBILITY for bipartite graphs on n vertices that is faster than O*(1.7804k) time?

References