Chordless Paths Through Three Vertices

Robert Haas[1] and Michael Hoffmann[2]

[1] IBM Zurich Research Laboratory, Rüschlikon
rha@zurich.ibm.com
[2] Institute for Theoretical Computer Science, ETH Zürich
hoffmann@inf.ethz.ch

Abstract. Consider the following problem, that we call "Chordless Path through Three Vertices" or CP3V, for short: Given a simple undirected graph $G = (V, E)$, a positive integer k, and three distinct vertices s, t, and $v \in V$, is there a chordless path from s via v to t in G that consists of at most k vertices? In a chordless path, no two vertices are connected by an edge that is not in the path. Alternatively, one could say that the subgraph induced by the vertex set of the path in G is the path itself. The problem has been raised in the context of service deployment in communication networks. We resolve the parametric complexity of CP3V by proving it $W[1]$-complete with respect to its natural parameter k. Our reduction extends to a number of related problems about chordless paths. In particular, deciding on the existence of a single directed chordless (s, t)-path in a digraph is also $W[1]$-complete with respect to the length of the path.

Keywords: graph theory, induced path, parameterized complexity.

1 Introduction

The number of specialized functions such as support for quality of service and protection against denial-of-service attacks, that is being built into network nodes is growing continuously. Thus it is becoming increasingly difficult for network administrators to use such sophisticated capabilities fully, especially when new services must be deployed in a timely manner. The advent of reprogrammable network nodes, made possible by cost-efficient network processors, has aggravated this issue as new capabilities may be introduced into the network dynamically in order to provision a particular service.

An automated method to perform service deployment was presented in [20], and several categories of services with similar deployment needs were introduced. This paper focuses on chordless paths: these are paths for which no two nodes are directly connected except along the path. Chordless paths are particularly relevant to the category of path-based services, i.e., services that require each node of a path from a source to a destination to be enabled with a common function. During the deployment of path-based services, nodes are queried for specific service requirements to determine whether they have the necessary capabilities to support a certain function. Such requirements are specific to each service, so that it is not recommended to let nodes advertise all their capabilities by default, which may even vary over time. Instead, on-demand query of relevant nodes is the preferred alternative.

Often it is necessary to query only a subset of the network nodes to determine whether a path with the required capabilities exists. If such a path contains a chord, there is a shorter path through the same set of nodes for this particular source/destination pair. Thus, for this type of query we are interested in only chordless paths. In particular, nodes that do not belong to any chordless path for a given source/destination pair are irrelevant and do not have to be queried.

It has been shown that for typical Internet router-level topologies, a large fraction of nodes do not belong to chordless paths [20]. To avoid unnecessary queries, it would hence be desirable to decide the following problem efficiently.

Problem 1 (CP3V). *Given an undirected graph $G = (V, E)$, a positive integer k, and three distinct vertices $s, t, v \in V$, is there a chordless path from s via v to t in G that consists of at most k vertices?*

But as we will prove in Section 4, this problem is $W[1]$-hard with respect to k; that is—roughly speaking—it is unlikely that an algorithm exists whose complexity is bounded by an arbitrary (say, doubly exponential) function in k but polynomial in the size of the input graph. As another consequence [1, 8], there is probably no PTAS to compute an $(1 + \varepsilon)$-approximation for the shortest chordless (s, v, t)-path in time bounded by an arbitrary function in ε but polynomial in the size of the input graph.

After summarizing some related results, we complement the above-mentioned hardness claim in Section 3 by proving CP3V to be in $W[1]$, that is, altogether $W[1]$-complete. Finally, Section 5 discusses how to extend our results for CP3V to a number of problems concerning chordless paths in directed graphs.

Related Work The following problem is a slight generalization of CP3V.

Problem 2 (Many Chordless (s, t)-Paths). *Given a simple undirected graph $G = (V, E)$, positive integers k and ℓ, and two distinct vertices $s, t \in V$, is there a set $U \subseteq V$ of at most k vertices such that the subgraph induced by U in G is a disjoint union of ℓ chordless (s, t)-paths?*

This was shown to be NP-complete by Fellows [15], already for $\ell = 2$, where it asks for a chordless cycle of length at most k through s and t. Let us refer to this problem as CC2V. As any chordless cycle through s has to pass through one of the neighbors of s, an instance of CC2V can be solved by less than $|V|$ calls to an algorithm that solves CP3V. In particular, this implies NP-completeness of CP3V. On the other hand, any CP3V instance can be converted to a CC2V instance in constant time: add a new vertex u to G and connect it to both s and t; now ask for a cycle through u and v of length at most $k + 1$. Hence, our $W[1]$-completeness result for CP3V also implies that CC2V is $W[1]$-complete.

The hardness results for CP3V and CC2V rely on graphs that contain many vertex-disjoint (s, t)-paths. It is not difficult to see that in planar graphs the existence of four vertex-disjoint (s, t)-paths basically[1] implies the existence of a chordless (s, v, t)-path. While this argument immediately gives an $\mathcal{O}(3^k n^c)$ time algorithm for CP3V (for some constant $c \in \mathbb{N}$), the more interesting question is whether Problem 2 is polynomial

[1] Except for some trivial cases which can be easily sorted out.

for planar graphs. This was answered in the affirmative for every fixed ℓ by McDiarmid et al. [23, 24].

If in Problem 2 we ask for vertex-disjoint paths only instead of requiring all paths to be jointly chordless, the problem is polynomial for general graphs and every fixed ℓ, even for arbitrary source-target pairs (s_i, t_i), $1 \leq i \leq \ell$ [27]. But it remains NP-complete if ℓ is considered part of the input [21].

Deciding whether a graph contains a chordless path of length at least k is one of the classical NP-complete problems (GT23 in [19]). Bienstock [3, 4] listed several other NP-complete problems related to chordless paths:

– Does a graph contain a chordless path of odd length between two specified vertices?
– Does a graph contain a chordless cycle of odd length (> 3) through a specified vertex?
– Does a graph contain a chordless path of odd length between any two vertices?

Note that these results do not imply the hardness of deciding whether there exists **any** path/cycle of odd length in a graph. This question is still open, see the discussion below.

Chordless cycles of length at least four are also called *holes*. They are tightly connected to Berge's strong perfect graph conjecture [2], whose proof has recently been announced by Chudnovsky et al. [9]. According to this conjecture, a graph is perfect[2] iff it is Berge, that is, if it contains neither an odd hole nor the complement of an odd hole. Hence, a polynomial time algorithm to decide whether there exists any odd hole in a given graph would immediately imply that perfect graphs can be recognized in polynomial time. Interestingly, no such an algorithm is known, although there are polynomial time algorithms [11, 10] to decide whether a graph is Berge, even independent of the strong perfect graph conjecture. Also, if the restriction to an odd number of vertices is omitted, the presence of holes can be detected in polynomial time: for holes on at least four vertices this is the well-studied recognition problem for chordal graphs [22, 28]. The problem of detecting holes on at least five vertices has recently been addressed by Nikolopoulos and Palios [26].

2 Notation

For a graph G denote by $V(G)$ the set of vertices in G, and denote by $E(G)$ the set of edges in G. For a vertex $v \in V$, denote by $N_G(v)$ the *neighborhood* of v in G, that is, the set of vertices from V that are adjacent to v in G. Similarly, for a set $W \subset V$ of vertices define $N_G(W) := \bigcup_{w \in W} N_G(w)$. The subscript is often omitted when it is clear which graph it refers to. A set $I \subset V$ of vertices is an *independent set* in G if E does not contain edges between any two vertices of I. For a set $W \subset V$ of vertices denote by $G[W]$ the *induced subgraph* of W in G, that is, the graph $(W, E \cap \binom{W}{2})$ or $(W, E \cap W^2)$ in the case of an undirected or a directed graph G, respectively.

A subgraph of G that has the form $(\{v_1, \ldots, v_k\}, \{\{v_i, v_{i+1}\} \mid 1 \leq i < k\})$ is called *path of length* k in G. Note that by the set notation we imply that $v_i \neq v_j$,

[2] In a perfect graph, the maximum number of pairwise adjacent vertices (*clique number*) for each induced subgraph is equal to the minimum number of colors needed to color the vertices in such a way that any two adjacent vertices receive distinct colors (*chromatic number*).

for $1 \leq i, j \leq k$ and that the length of a path is defined as its number of vertices. The vertices v_1 and v_k are referred to as the path's *endpoints*, the other vertices are called *interior*. Two paths are called *vertex-disjoint* iff they do not share vertices except for possibly common endpoints. For two vertices $s, t \in V$ any path from s to t in G is called (s, t)-path. More generally, if vertices s, v, and t appear on path P in this order, we call P an (s, v, t)-path. A path P in G is called *chordless* if $V(P)$ is an independent set in $(V, E \setminus E(P))$. An alternative equivalent definition would be to call a path P in G chordless iff $G[V(P)] = P$. Hence, such paths are also known as *induced paths*.

Parameterized Complexity To cope with the apparent computational intractability of NP-hard problems, attempts were made to analyze more closely which parts or aspects of the input render a particular problem hard. A prototypical example is *Vertex Cover*, which asks for a set C of at most k vertices from a given graph G on n vertices such that for each edge at least one endpoint is in C. The trivial observation that for any edge at least one of the two incident vertices has to be in C, leads to an $O(2^k n)$ time algorithm: choose an arbitrary edge and branch on the two possibilities, in both cases removing one vertex and all incident edges from the graph. Hence, the intractability of Vertex Cover is connected to the number k of vertices in the cover rather than to the size of the graph G. One says that Vertex Cover is *fixed-parameter-tractable* (FPT) with respect to the parameter k because there is an algorithm that runs in $O(f(k)p(n))$ for an arbitrary, typically exponential, function f and a polynomial function p.

Naturally, there are also problems for which it is not known whether their complexity can be isolated into a particular parameter in this way. Moreover, similar to the classical complexity classes, there are classes of parameterized problems that are hard in the sense that if there is an FP algorithm for any of them, then all of them are FPT. The most important such class is called $W[1]$, which can be described in terms of the following "canonical" problem.

Problem 3 (Weighted q-CNF-Satisfiability). *Given positive integers q and k, and a boolean formula F in conjunctive normal form such that each clause contains at most q literals, is there a satisfying assignment for F with at most k variables set to true?*

A problem P parameterized by k is said to be *m-reducible* to a problem P' parameterized by k' iff there is a function h that maps an instance (x, k) of P to an instance (x', k') of P' such that $k' = g(k)$ and x' can be computed in time $f(k)p(x)$, for arbitrary functions f and g, and a polynomial function p. Now $W[1]$ is defined as the class of parameterized problems that can be m-reduced to Weighted q-CNF-Satisfiability for some constant q. Finally, a problem is $W[1]$-*hard* iff every problem in $W[1]$ can be m-reduced to it. A problem that is both $W[1]$-hard and in $W[1]$ is called $W[1]$-*complete*.

At this point, we refer the interested reader to the literature for more in-depth information about parameterized complexity. The book of Downey and Fellows [13] provides a thorough treatment of complexity-theoretic aspects, whereas the survey of Niedermeier [25] focuses more on algorithms.

3 Membership in W[1]

In this section we analyze the parameterized complexity of CP3V and prove the problem to be in $W[1]$ with respect to its natural parameter, the path length. As a first step, note that a chordless (s, t)-path P is already determined by its set of vertices. For example, only one neighbor x of s can be in $V(P)$ because any later visit of another neighbor would introduce a chord. Similarly, exactly one neighbor of x (other than s) can be in $V(P)$. In this manner P can be uniquely reconstructed from $V(P)$.

Proposition 4. *A subgraph P of G is a chordless (s, t)-path if and only if P is connected, s and t have degree one in $G[V(P)]$, and all vertices other than s and t have degree two in $G[V(P)]$.* □

We do not know how to reduce CP3V to Weighted q-CNF-Satisfiability. Instead, we reduce to a different problem called SNTMC that is defined below. SNTMC is known to be $W[1]$-complete [5, 14], and reduction to SNTMC and its relatives has proven to be a useful tool to establish membership results within the W-hierarchy [6, 7].

Problem 5 (Short Nondeterministic Turing Machine Computation (SNTMC)). *Given a single-tape, single-head nondeterministic Turing machine M, a word x on the alphabet of M, and a positive integer k, is there a computation of M on input x that reaches a final accepting state in at most k steps?*

Theorem 6. CP3V *is in $W[1]$ w.r.t. k.*

Proof. Consider an instance (G, s, v, t, k) of CP3V, where $G = (V, E)$ is a simple undirected graph, $s, v, t \in V$, and k is a positive integer. We will construct an instance (M, k') of SNTMC such that there is a computation for M that reaches a final accepting state in at most $k' = k^2 + 3k - 1$ steps iff there exists a chordless (s, v, t)-path of length at most k in G. (It is important that k' depends on k only and not on n.) A schematic view of the construction is shown in Fig. 1.

Let $M = (\Sigma, Q, \Delta, g_1, \{A\})$, where the alphabet Σ is defined as $\Sigma := \{\Box\} \cup \{\sigma_u \mid u \in V\}$; the state set is

$$Q := \{A, R\} \cup \{g_i \mid 1 \le i \le k\} \cup \{a, b, c, d, l, r\} \cup \{p_u \mid u \in V\} \cup \{q_u \mid u \in V\} ;$$

the transition relation $\Delta : Q \times \Sigma \times Q \times \Sigma \times \{+, -, 0\}$ is defined below; the initial state is g_1; the final accepting state is A, and the final rejecting state is R. When the Turing machine starts, all tape cells contain the blank symbol ($\Box$). The computation consists of three phases: first, the at most k vertices of a chordless (s, v, t)-path P in G are “guessed” by writing the sequence of corresponding symbols σ_u, $u \in V(P)$, onto the tape. The next two phases are completely deterministic and check that P visits s, v, and t in the order given, and that P is a chordless path in G.

First Phase: The Turing machine may write up to k arbitrary vertex symbols onto the tape: $(g_i, \Box, g_{i+1}, \sigma_u, +) \in \Delta$, for all $u \in V$ and all $1 \le i < k$, and $(g_i, \Box, a, \sigma_u, 0) \in \Delta$, for all $u \in V$ and all $1 \le i \le k$. (The transition specifies, in order, current state, symbol under the head, new state after transition, symbol to write to the tape, and movement of the head: $+$ for right, $-$ for left, and 0 for stay.) After the first phase, the Turing machine is in state a and the sequence of between one and k vertex symbols starts at the current tape cell, extending to the left.

Second Phase: Check whether the guessed sequence visits t, v, and s, in order. The rightmost symbol should be σ_t: $(a, \sigma_t, b, \sigma_t, -) \in \Delta$. Then somewhere σ_v must appear: $(b, \sigma_u, b, \sigma_u, -) \in \Delta$, for all $u \in V \setminus \{v\}$, and $(b, \sigma_v, c, \sigma_v, -) \in \Delta$. The final symbol has to be σ_s: $(c, \sigma_u, c, \sigma_u, -) \in \Delta$, for all $u \in V \setminus \{s\}$, and $(c, \sigma_s, d, \sigma_s, -) \in \Delta$. Nothing may follow after s: $(d, \square, l, \square, +) \in \Delta$. For all state/symbol combinations that are not explicitly mentioned (for example, $(b, \square)$ or (a, σ_v)) there is a transition to the final rejecting state R. After the second phase, the machine is in state l and the head points towards the leftmost of the symbols that have been guessed in Phase 1. The content of the tape remains unchanged during Phase 2.

Third Phase: Scan and remove the first vertex: $(l, \sigma_u, p_u, \square, +) \in \Delta$, for all $u \in V$. If no more vertex is left at this point, we are done: $(p_u, \square, A, \square, 0) \in \Delta$, for all $u \in V$. Otherwise, the next vertex should be adjacent: $(p_u, \sigma_w, q_u, \sigma_w, +) \in \Delta$, for all $u, w \in V$ for which $\{u, w\} \in E$. Whatever follows must not be adjacent: $(q_u, \sigma_w, q_u, \sigma_w, +) \in \Delta$, for all $u, w \in V$ with $u \neq w$ and $\{u, w\} \notin E$. If all vertices have been checked, return to the leftmost: $(q_u, \square, r, \square, -) \in \Delta$, for all $u \in V$, and $(r, \sigma_w, r, \sigma_w, -) \in \Delta$, for all $w \in V$. Finally, re-iterate: $(r, \square, l, \square, +) \in \Delta$. Again, all state/symbol combinations that are not explicitly mentioned lead to the final rejecting state R. Note that after the third phase, all tape cells contain the blank symbol again.

Phase 3 ensures that all vertices guessed in Phase 1 are distinct, as, otherwise the right scan in state q_u, for some $u \in V$, fails. Moreover, because of the transition from p_u to q_u, the vertices chosen form a path P in G. The right scan in state q_u also ensures that no two of the vertices are connected except along P. Finally, in Phase 2 we check that the endpoints of P are s and t, and that P visits v. Altogether, the machine reaches an accepting state iff it guesses the vertices of a chordless (s, v, t)-path of length at most k in Phase 1. An easy calculation reveals that if k symbols are written onto the tape in Phase 1, the remaining computation consists of exactly $k^2 + 2k - 1$ transitions. □

4 Hardness for $W[1]$

In this section, we prove that CP3V is $W[1]$-hard using a reduction from Independent Set, which is one of the "classical" $W[1]$-hard problems [12].

Problem 7 (Independent Set). *Given a simple undirected graph $G = (V, E)$ and a positive integer k, is there an independent set of size at least k in G?*

Consider an instance of Independent Set, that is, a graph $G = (V, E)$ and an integer k, $1 \leq k \leq |V|$, and let $V = \{v_1, \ldots, v_n\}$. We construct a graph G' from G such that the answer to the CP3V problem on G' provides the solution to the independent-set problem on G.

The main ingredient for our construction is called *vertex choice diamond*: it consists of n vertices $v_1^i, \ldots, v_n^i$ plus two extra vertices s^i and t^i connected to each of the n vertices v_j^i, $1 \leq j \leq n$, as shown in Fig. 2. Clearly, there are exactly n chordless (s^i, t^i)-paths in such a diamond. As the naming of the vertices suggests, we associate each of these paths with a vertex from G in a bijective manner: routing a path through v_j^i, for some $1 \leq j \leq n$, is interpreted as selecting v_j to be part of the independent set

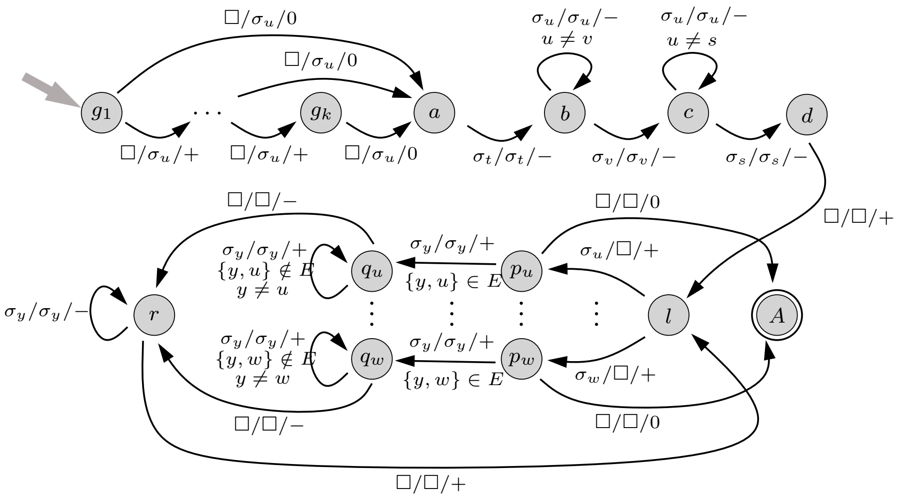


Fig. 1. A schematic description of the Turing machine defined in the proof of Theorem 6. The transition arrows are labeled by, in order, symbol under head, symbol to write, and head movement. To increase readability the final rejecting state and all transitions to it have been omitted.

I to be constructed. The construction uses k such vertex choice diamonds, which are connected by identifying s^{i+1} and t^i, for all $1 \leq i < k$. Let us call the graph described so far G_{VC}, where VC stands for vertex choice.

Proposition 8. *Any chordless (s^1, t^k)-path of length ℓ in G_{VC} corresponds to a set of $\ell - k - 1$ vertices in G.* □

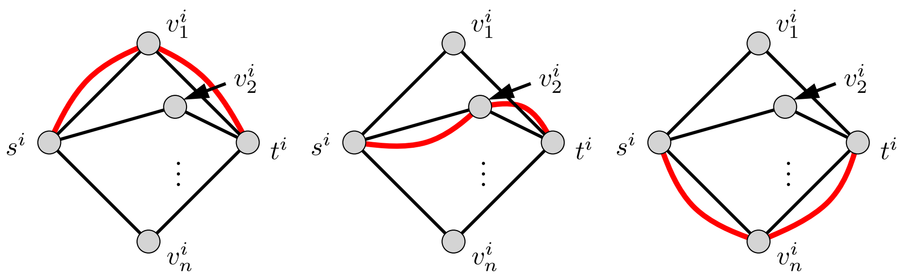
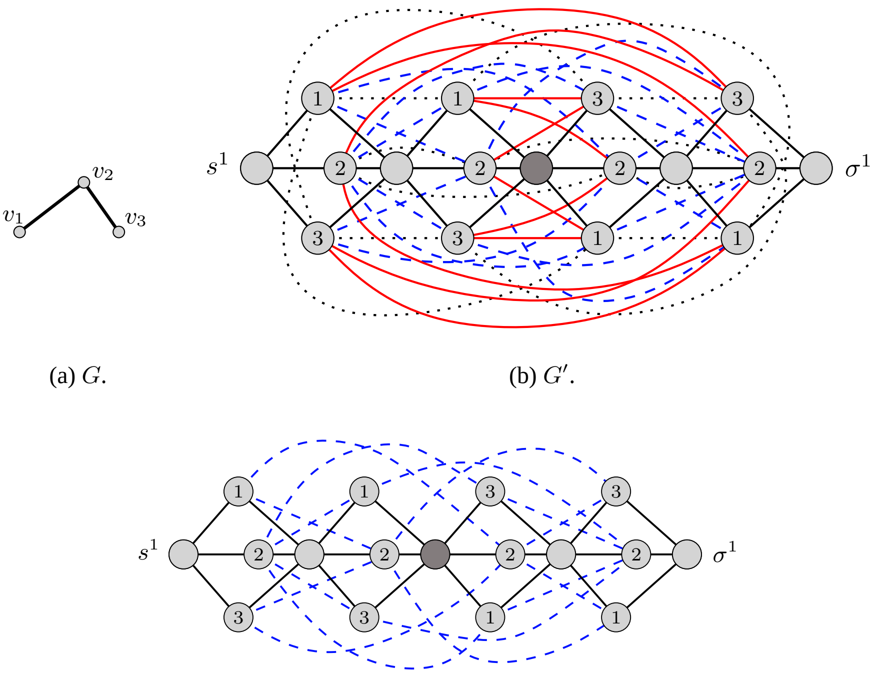


Fig. 2. A vertex choice diamond and three of its n chordless (s^i, t^i)-paths.

The next step is to ensure that the vertex sets chosen by traversing G_{VC} on a chordless path correspond to independent sets in the original graph G. To accomplish this, we construct G' from two symmetric copies of G_{VC}. Denote the vertices in the first copy C of G_{VC} by s^i, v_j^i, and t^i, whereas the vertices in the second copy Γ of G_{VC} are referred to as σ^i, φ_j^i, and τ^i, for $1 \leq i \leq k$ and $1 \leq j \leq n$. The graphs C and Γ are connected by identifying t^k and τ^k. The construction of G' is completed by adding a number of edges that encode the adjacency of G. An example is shown in Fig. 3.

- There is an edge in G' between v_j^i and φ_ℓ^i, for all $1 \leq i \leq k$ and all $1 \leq j, \ell \leq n$ with $j \neq \ell$. Such an edge is called *consistency edge*.
- For every edge $\{v_p, v_q\}$ in G, connect the vertex sets $\{v_p^i, \varphi_p^i\}$ and $\{v_q^j, \varphi_q^j\}$, for all $1 \leq i, j \leq k$ with $i \neq j$, by a complete bipartite subgraph in G'. These edges are called *independence edges*.
- The vertices v_ℓ^i and φ_ℓ^i are connected by an edge in G' to all of the vertices v_ℓ^j and φ_ℓ^j, for all $1 \leq i, j \leq k$ with $i \neq j$. These edges are called *set edges*.



(a) G.

(b) G'.

(c) G' (independence edges only).

Fig. 3. An example illustrating the construction of G' for $k = 2$. Consistency edges are shown by solid lines, independence edges by dashed lines, and set edges by dotted lines. The vertex labels in G' indicate the correspondence to the vertices from G: for example, the vertices labeled "1" correspond to v_1. The vertex $t^k = \tau^k$ is shaded dark.

Lemma 9. *No chordless (s^1, σ^1)-path via t^k in G' uses a consistency edge or an independence edge or a set edge.*

Proof. Let P be a chordless (s^1, σ^1)-path via t^k in G'. Consider the initial part of P which traverses (part of) the first vertex choice diamond of C. By construction of G', exactly one of the vertices v_j^1, $1 \leq j \leq n$ is on P. (All of these vertices are neighbors of s^1.) Similarly, the final part of P contains exactly one of the vertices φ_ℓ^1, $1 \leq \ell \leq n$.

Moreover, if $\ell \neq j$ vertices v_j^1 and φ_ℓ^1 are connected by an edge in G'. Hence, they together with s^1 and σ^1 induce an (s^1, σ^1)-path in G' that does not visit v and cannot be extended to a chordless path visiting v either. Thus, we may conclude that $\ell = j$.

Furthermore, note that by construction v_j^1 and φ_j^1 have the same neighbors along both independence and set edges. Thus, if P continues along an independence edge or set edge from either vertex, it has to do so from the other vertex as well, as otherwise the unused edge would form a chord, but, as above, P cannot reach t^k in this case.

In summary, the initial part of P goes from s^1 via v_j^1, for some $1 \leq j \leq n$, to $t^1 = s^2$, and the final part of P is completely symmetric: from τ^1 via φ_j^1 to σ^1. By induction on k, the initial part of P is an (s^1, t^k)-path Q that visits all s^i in increasing order, for $1 \leq i \leq n$, without using any consistency, independence, or set edge, and the final part of P is a (τ^k, σ^1)-path that is completely symmetric to Q. □

Theorem 10. CP3V *is* $W[1]$*-hard w.r.t.* k.

Proof. Given an instance (G, k) of Independent Set, we construct the graph G' as described above. The graph G' contains $2k(n+1)+1$ vertices. To compute the number of edges in G' note that there are $4kn$ edges in the $2k$ vertex choice diamonds, plus $kn(n-1)$ consistency edges, $4mk(k-1)$ independence edges, and $2nk(k-1)$ set edges, where $n := |V(G)|$ and $m := |E(G)|$. Hence, G' can be constructed from G in time and space polynomial in both n and k.

Let P be a chordless (s^1, σ^1)-path via t^k of length at most $4k+1$ in G'. By Lemma 9, P has a very special form: in particular, its length is exactly $4k+1$, and it visits exactly one vertex $v_{j_i}^i$ from each of the vertex choice diamonds with $1 \leq i \leq n$. Let $I := \{v_{j_i} \in V(G) \mid v_{j_i}^i \in V(P) \text{ for } 1 \leq i \leq n\}$. Suppose that $v_{j_i}^i = v_{j_\ell}^\ell$ for some $1 \leq i, \ell \leq k$ with $i \neq \ell$. As $v_{j_i}^i$ and $v_{j_\ell}^\ell$ are connected by a set edge in G' that is not in P by Lemma 9, this set edge forms a chord of P, in contradiction to our assumption that P is chordless. Therefore, the vertices $v_{j_i}^i$ visited by P correspond to mutually distinct vertices in G, that is, together with Proposition 8 it follows $|I| = 2k+1-k-1 = k$.

Furthermore, we claim that I is an independent set in G. Suppose that for two vertices $v_{j_i}^i$ and $v_{j_\ell}^\ell$ on P, $1 \leq i, \ell \leq k$ and $i \neq \ell$, the corresponding vertices $v_{j_i}^i$ and v_{j_ℓ} are neighbors in G. Then by construction $v_{j_i}^i$ and $v_{j_\ell}^\ell$ are connected by an independence edge in G'. Again, this edge is not in P by Lemma 9, that is, it forms a chord of P, in contradiction to our assumption that P is chordless. Therefore, no two vertices in I are adjacent in G.

Conversely, it is easy to see that for any independent set $I = \{v_1, v_2, \ldots, v_k, \ldots\}$ (without loss of generality) of size at least k in G there is a chordless (s^1, σ^1)-path P via t^k of length $4k+1$ in G': in the i-th vertex choice diamonds, P visits v_i^i and φ_i^i, for $1 \leq i \leq k$.

Therefore, we have a parameterized reduction from an independent set instance (G, k) to a CP3V instance $(G', 4k+1)$, establishing $W[1]$-hardness of CP3V. □

5 Chordless Paths in Digraphs

The notion of chordless paths generalizes in a straightforward manner to digraphs.

Problem 11 (Directed Chordless (s,t)-Path (DCP)). *Given a simple digraph $G = (V, E)$, a positive integer k, and two distinct vertices $s, t \in V$, is there a chordless directed (s,t)-path of length at most k in G?*

Fellows et al. [16] showed that DCP is NP-complete even if restricted to planar digraphs. Our constructions described above can easily be adapted to the directed setting.

Theorem 12. DCP *is $W[1]$-complete w.r.t. k.*

Proof. In the Turing machine of Theorem 6 replace all conditions that require the existence of an edge by corresponding conditions requiring the presence of a directed edge. Similarly, all conditions requiring the absence of an edge are replaced by corresponding conditions disallowing both directed edges.

The construction described in Theorem 10 is modified as follows. In the vertex choice diamonds direct all edges from s^i to v^i_j and from v^i_j to t^i, for all $1 \le i \le k$ and all $1 \le j \le n$. In the symmetric copy, direct all edges from τ^i to φ^i_j and from φ^i_j to σ^i, for all $1 \le i \le k$ and all $1 \le j \le n$. These orientations induce a linear ordering $(s^1, \ldots, t^k = \tau^k, \ldots, \sigma^1)$ on $V(G')$. The remaining edges, that is, the consistency, independence, and set edges all are oriented from the vertex that is greater with respect to this linear order to the smaller vertex. It is easy to verify that no chordless directed (s^1, σ^1)-path can use a consistency, independence, or set edge. In fact, the orientation is chosen such that any chordless directed (s^1, σ^1)-path in G' passes through $t^k = \tau^k$, although this is not required by definition, in contrast to CP3V. □

As a consequence, also the following problem is $W[1]$-complete w.r.t. k. (Just add a single directed edge (σ^1, s^1) to the construction described in Theorem 12.)

Problem 13 (Directed Chordless Cycle). *Given a simple digraph $G = (V, E)$, a positive integer k, and a vertex $s \in V$, is there a chordless directed cycle of length at most k through s in G?*

Note that both problems are polynomial if the path or cycle is not required to be chordless: the maximum number k of vertex-disjoint directed (s,t)-paths can be computed in $O(k|E|)$ time using flow techniques [17]. However, deciding whether there exist a directed (s_1, t_1)-path and a directed (s_2, t_2)-path that are vertex-disjoint is NP-complete, even for $t_1 = s_2$ and $t_2 = s_1$ [18].

Also, if the definition of chordless is relaxed to allow "back-cutting" arcs within each path, DCP restricted to planar graphs is polynomial, even for an arbitrary but fixed number of chordless (s,t)-paths [24]. The existence of such arcs is the crucial difference between the directed and the undirected problem: in an undirected (s,t)-path P every edge joining two vertices that are non-adjacent along P can be used as a shortcut. That is, the presence of any (s,t)-path implies the existence of a chordless (s,t)-path. However, we will show below that admitting back-cutting arcs does not change the parametric complexity of the problem for general graphs.

Definition 14. *An (s,t)-path P in a graph $G = (V, E)$ is called* ***weakly chordless*** *iff P is a shortest (s,t)-path in $G[V(P)]$.*

Observe that there is no difference between chordless and weakly chordless in undirected graphs. But, in contrast to DCP, the presence of a directed weakly chordless (s,t)-path can be decided in linear time by a breadth-first search. However, the generalization to several paths defined below is again $W[1]$-complete, already for two paths.

Problem 15 (Many Weakly Chordless (s,t)-Paths). *Given a simple digraph $G = (V,E)$, positive integers k and ℓ, and two distinct vertices $s,t \in V$, is there a set $U \subseteq V$ with $|U| \leq k$ such that $G[U]$ is a disjoint union of ℓ weakly chordless (s,t)-paths?*

Theorem 16. *Two Weakly Chordless (s,t)-Paths is $W[1]$-complete w.r.t. k.*

Proof. It is clear how to adapt the Turing machine construction of Theorem 6 to establish membership in $W[1]$.

The construction described in Theorem 10 is modified as follows. First, add a directed edge from σ^1 to s^1, and let $s := t^k = \tau^k$ and $t := s^k$. In the vertex choice diamonds direct all edges from t^i to v_j^i and from v_j^i to s^i, for all $1 \leq i \leq k$ and all $1 \leq j \leq n$. Likewise, in the symmetric copy direct all edges from τ^i to φ_j^i and from φ_j^i to σ^i, for all $1 \leq i \leq k$ and all $1 \leq j \leq n$. Remove all independence and set edges within the same diamond chain, such that all remaining independence or set edges are between v_j^i and φ_q^p, for some $1 \leq i,p \leq k$ and $1 \leq j,q \leq n$. Direct those edges from v_j^i towards φ_q^p.

Consider (t^k, s^1)-paths P and Q in G' such that $G[V(P) \cup V(Q)]$ is a disjoint union of weakly chordless (t^k, s^1)-paths. The way the edges are directed, one of the paths, say, P comes via the vertex choice diamonds and visits the vertices $t^k, s^k, t^{k-1}, \dots t^1, s^1$, in order. On the other hand, Q traverses the symmetric copy and visits the vertices $\tau^k, \sigma^k, \tau^{k-1}, \dots \tau^1, \sigma^1$, in order, before finally reaching s^1 via the added edge. Because there must not be any edge between P and Q, we can argue as in Theorem 10 that the vertices v_j^i and φ_q^p visited by P and Q, respectively, correspond to an independent set of size at most k in G. $\square$

Acknowledgments We thank Yoshio Okamoto and Emo Welzl for helpful discussions.

References

1. BAZGAN, C. Schémas d'approximation et complexité paramétrée. Rapport du stage (DEA), Université Paris Sud, 1995.
2. BERGE, C. Färbung von Graphen deren sämtliche beziehungsweise deren ungerade Kreise starr sind (Zusammenfassung). *Wiss. Z. Martin Luther Univ. Halle Wittenberg Math. Naturwiss. Reihe* (1961), 114–115.
3. BIENSTOCK, D. On the complexity of testing for odd holes and induces odd paths. *Discrete Math. 90*, 1 (1991), 85–92.
4. BIENSTOCK, D. Corrigendum to: On the complexity of testing for odd holes and induces odd paths. *Discrete Math. 102*, 1 (1992), 109.
5. CAI, L., CHEN, J., DOWNEY, R. G., AND FELLOWS, M. R. On the parameterized complexity of short computation and factorization. *Arch. Math. Logic 36*, 4–5 (1997), 321–337.
6. CESATI, M. Perfect code is W[1]-complete. *Inform. Process. Lett. 81*, 3 (2002), 163–168.

7. CESATI, M. The Turing way to parameterized complexity. *J. Comput. Syst. Sci. 67*, 4 (2003), 654–685.
8. CESATI, M., AND TREVISAN, L. On the efficiency of polynomial time approximation schemes. *Inform. Process. Lett. 64*, 4 (1997), 165–171.
9. CHUDNOVSKY, M., ROBERTSON, N., SEYMOUR, P. D., AND THOMAS, R. The strong perfect graph theorem. Manuscript (2003).
10. CHUDNOVSKY, M., AND SEYMOUR, P. D. Recognizing Berge graphs. Manuscript (2003).
11. CORNUÉJOLS, G., LIU, X., AND VUŠKOVIĆ, K. A polynomial algorithm for recognizing perfect graphs. In *Proc. 44th Annu. IEEE Sympos. Found. Comput. Sci.* (2003), pp. 20–27.
12. DOWNEY, R. G., AND FELLOWS, M. R. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoret. Comput. Sci. 141* (1995), 109–131.
13. DOWNEY, R. G., AND FELLOWS, M. R. *Parameterized Complexity*. Monographs in Computer Science. Springer-Verlag, 1999.
14. DOWNEY, R. G., FELLOWS, M. R., KAPRON, B., HALLETT, M. T., AND WAREHAM, H. T. Parameterized complexity and some problems in logic and linguistics. In *Proc. 2nd Workshop on Structural Complexity and Recursion-theoretic methods in Logic-Programming* (1994), vol. 813 of *Lecture Notes Comput. Sci.*, Springer-Verlag, pp. 89–101.
15. FELLOWS, M. R. The Robertson-Seymour theorems: A survey of applications. In *Proc. AMS-IMS-SIAM Joint Summer Research Conf.*, Providence, RI, 1989, pp. 1–18.
16. FELLOWS, M. R., KRATOCHVÍL, J., MIDDENDORF, M., AND PFEIFFER, F. The complexity of induced minors and related problems. *Algorithmica 13* (1995), 266–282.
17. FORD, L. R., AND FULKERSON, D. R. Maximal flow through a network. *Canad. J. Math. 8* (1956), 399–404.
18. FORTUNE, S., HOPCROFT, J. E., AND WYLLIE, J. The directed subgraph homeomorphism problem. *Theoret. Comput. Sci. 10* (1980), 111–121.
19. GAREY, M. R., AND JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, 1979.
20. HAAS, R. *Service Deployment in Programmable Networks*. PhD thesis, ETH Zurich, Switzerland, 2003.
21. KARP, R. M. On the complexity of combinatorial problems. *Networks 5* (1975), 45–68.
22. LUEKER, G. S., ROSE, D. J., AND TARJAN, R. E. Algorithmic aspects of vertex elimination in graphs. *SIAM J. Comput. 5* (1976), 266–283.
23. MCDIARMID, C., REED, B., SCHRIJVER, A., AND SHEPHERD, B. Non-interfering network flows. In *Proc. 3rd Scand. Workshop Algorithm Theory* (1992), pp. 245–257.
24. MCDIARMID, C., REED, B., SCHRIJVER, A., AND SHEPHERD, B. Induced circuits in planar graphs. *J. Combin. Theory Ser. B 60* (1994), 169–176.
25. NIEDERMEIER, R. *Invitation to Fixed-Parameter Algorithms*. Habilitation thesis, Wilhelm-Schickard Institut für Informatik, Universität Tübingen, Germany, 2002.
26. NIKOLOPOULOS, S. D., AND PALIOS, L. Hole and antihole detection in graphs. In *Proc. 15th ACM-SIAM Sympos. Discrete Algorithms* (2004), pp. 843–852.
27. ROBERTSON, N., AND SEYMOUR, P. D. Graph minors XIII. The disjoint paths problem. *J. Combin. Theory Ser. B 63* (1995), 65–110.
28. TARJAN, R. E., AND YANNAKAKIS, M. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput. 13* (1984), 566–579.