On the treewidth of dynamic graphs

Bernard Mans, Luke Mathieson

Department of Computing, Macquarie University, Sydney, NSW 2109, Australia

ARTICLE INFO
Available online 8 January 2014

Keywords:
Dynamic graphs
Metatheorems
Treewidth
Local treewidth
Bounded expansion

ABSTRACT

Dynamic graph theory is a novel, growing area that deals with graphs that change over time and is of great utility in modeling modern wireless, mobile and dynamic environments. As a graph evolves, possibly arbitrarily, it is challenging to identify the graph properties that can be preserved over time and understand their respective computability. In this paper we are concerned with the treewidth of dynamic graphs. We focus on metatheorems, which allow the generation of a series of results based on general properties of classes of structures. In graph theory two major metatheorems on treewidth provide complexity classifications by employing structural graph measures and finite model theory. Courcelle's Theorem gives a general tractability result for problems expressible in monadic second order logic on graphs of bounded treewidth, and Frick and Grohe demonstrate a similar result for first order logic and graphs of bounded local treewidth. We extend these theorems by showing that dynamic graphs of bounded (local) treewidth where the length of time over which the graph evolves and is observed is finite and bounded can be modeled in such a way that the (local) treewidth of the underlying graph is maintained. We show the application of these results to problems in dynamic graph theory and dynamic extensions to static problems. In addition we demonstrate that certain widely used dynamic graph classes have bounded local treewidth.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Graph theory has proven to be an extremely useful tool for modeling computational systems and with the advent and growing preponderance of mobile devices and dynamic systems it is natural that graph theory is extended and adapted to capture the evolving aspects of these environments. Dynamic graphs have been formalized in a number of ways: e.g., time-varying graphs [7,8,20], carrier-based networks [4], evolving graphs [6,19,5], delay-tolerant networks [26], dynamic networks [29,28], scheduled networks [1], temporal networks [27], opportunistic networks [9,25], Markovian [10]. When considering the dynamic aspects of a dynamic graph, even classically simple properties such as shortest paths become more complex to compute and may even become definitionally ambiguous [5] (Fig. 1).

In this paper, we are not particularly interested in any particular dynamic model. Initially we will loosely use the term dynamic graph and for our purpose we will define the term formally using the simplest possible definition as it is a generalized and reasonably assumption free model for a dynamic graph. For this paper, one of the key questions in moving from static graph theory to dynamic graph theory concerns the preservation of properties and their computability.

One important general approach to the complexity and computability of properties in (static) graphs is the application of metatheorems which classify large classes of problems. An important metatheorem is Courcelle's Theorem (stated and

✩ This paper contains results published in preliminary form in [30].
* Corresponding author.
E-mail addresses: bernard.mans@mq.edu.au (B. Mans), luke.mathieson@mq.edu.au (L. Mathieson).
proved over a series of articles from [12] to [13]) which gives a polynomial time algorithm for any monadic second-order expressible property on graphs of bounded treewidth. More precisely the model checking problem for monadic second-order logic is fixed-parameter tractable with parameter \(|\varphi| + k\) where \(\varphi\) is the sentence of logic and \(k\) is the treewidth of the input structure. Frick and Grohe [23] give a similar metatheorem for first-order logic and structures of locally bounded treewidth, which allows a greater class of structures (all structures of bounded treewidth have locally bounded treewidth), but constrains the logical expressibility. In fact Dvořák et al. [16] show that properties expressible in first-order logic are decidable in linear time for graphs of bounded expansion, a superclass of several classes of sparse graphs, including those with bounded local treewidth. Stewart [34] demonstrates that Frick and Grohe’s result holds if the bound on the local treewidth is dependent on the parameter of the input problem, rather than simply being constant.

Such metatheorems are extremely useful classification tools, and having them available for use in the context of dynamic graphs would be highly desirable. Two questions immediately arise when considering such an extension — are there any restrictions that have to be made to the logic and are there any further constraints on the structures (in this case the graphs)? Answering the first question is simple: no. An apparent natural match for dynamic graphs is temporal logic (a form of modal logic), however temporal logic has a simple canonical translation into first-order and monadic second-order logics. The addition of a “time” relation, that provides a temporal ordering is sufficient to capture the ideas of temporal logic. Thus we are not limited (more than before) with the logics that are applicable. With regards to the structure, we must first consider the setting of these metatheorems. They are both true in the context of finite model theory, emphasizing the finite. Therefore we are immediately limited to finite temporal domains (a finite domain for the vertices and hence edges of the graph is expected of course). Furthermore the tractability results we derive (at least in some cases) rely on taking the size of the temporal domain as a parameter. For a number of interesting application areas of dynamic graphs this is quite natural: periodic and recurrent graphs (see [7] and [20] for details) intrinsically bounded temporal domains. In other cases where we have a total order on the temporal domain, the parameter dependence on the size of the temporal domain may be removed. However, as we require the length of the logical sentence to be bounded by some function of the parameter, we still encounter the usual problem with logically expressing a relation between elements that are an unbounded distance apart. Of course this is even a problem encountered when trying to express simple connectivity in first order logic. Beyond this if we are to capture the temporal aspect in the structure, we must do so in a manner that is both usable and respects the constraints on the structure necessary for the metatheorems — namely bounded treewidth, local treewidth and bounded expansion.

Dynamic problems have been approached before in a number of ways. For instance, Hagerup [24] examines the situation where the logical expression changes (i.e. the question being asked changes), and Bodlaender [3], Cohen et al. [11] and Frederickson [22] give a variety of results that deal with graphs that change, requiring changes in the tree decompositions used. In contrast this work deals with problems that include the temporal dimension as an intrinsic aspect, rather than problems that can be answered at an instant in time (and then may need to be recomputed). A simple example is a journey, the temporal extension of a path, where the route between two vertices may not exist at any given point of time, but as the graph changes the schedule of edges may allow traversal.

The remainder of this paper deals with the translation of dynamic graphs into structures that maintain these bounds if the original graph did so at every point in time. In addition we demonstrate the utility of these metatheorems in classification by application to some open problems on dynamic graphs.

2. Preliminaries

2.1. Graphs and dynamic graphs

The graphs we employ are simple graphs, both directed and undirected. For the usual graph properties we use the usual notation (qq.v. Diestel [14]). In particular, given a graph \(G\), we denote the number of vertices of \(G\) by \(|G|\) and the number of edges by \(|E(G)|\).
We define a dynamic graph for our purposes as a (di)graph augmented with a set \( T \) of times, a function \( \zeta_v \) which maps the edge set to a subset of \( T \) and a function \( \zeta_e \) that maps the vertex set to subsets of \( T \), representing the times at which the edge or vertex exists. We will consider a discrete and finite set \( T \) of times for two main reasons: (i) typically, in most applications at hand, a continuous set \( T \) could be easily made discrete when considering time of changes of the graph as an event-based model; (ii) similarly, applications of interest are not focused on the observation of the evolution of the graph over an infinite period, but rather on some sufficiently large, yet finite, period where some expected or asymptotic behavior can be identified.

More formally a dynamic graph \( G = (V,E,T,\zeta_v,\zeta_e) \) consists of a set \( V \) of vertices, a set \( E \subseteq V \times V \) of edges (where \( \forall v \notin E \) for any \( v \in V \)), a set \( T \) of times, a function \( \zeta_v : V \rightarrow 2^T \) and a function \( \zeta_e : E \rightarrow 2^T \).

We denote the static graph derived from a dynamic graph \( G \) by taking the snapshot at time \( t \) as \( G_t \). We do not allow edges to exist when either of their vertices do not, although the case where they do could be of interest in modeling, for example, wireless networks with long transmission ranges. We also assume that there is some order \( <_T \) defined over \( T(G) \), however except where specified we do not assume that this order has any particular properties (i.e. we assume neither totality nor transitivity).

A journey in a dynamic graph is simply a path over time, that is it is a path where the edges (and vertices) of the path may not all exist at a single point in time, but where there is an ordered sequence of times such that each edge in sequence exists at a point in time after its predecessor. Note that this inherently makes a journey directed. We will denote a journey from vertex \( u \) to vertex \( v \) by \( J(u,v) \).

This dynamic graph definition is independent from other referenced definitions, though relationships can be drawn. Again, here, we aim to focus on the graph theoretic aspects, rather than modeling a particular system, thus we attempt to keep the definition as simple and open as possible. We also note that a similar definition with a continuous notion of time can be converted to an event-based model, which then fulfills the requirements of our definition.

2.2. Treewidth, local treewidth and bounded expansion

A tree decomposition \( \mathcal{T} \) of a graph \( G \) is a tree \( \mathcal{T} \) and mapping \( \beta : V(\mathcal{T}) \rightarrow 2^{V(G)} \) with the following properties:

- \( \bigcup_{v \in V(\mathcal{T})} \beta(t) = V(G) \).
- For each edge \( uv \in E(G) \) there is some \( t \in V(\mathcal{T}) \) such that \( u, v \in \beta(t) \).
- For each \( v \in V(G) \), the set \( \{ t \in V(\mathcal{T}) \mid v \in \beta(t) \} \) of tree vertices induces a connected subgraph of \( \mathcal{T} \).

As is standard, we call \( \beta(t) \) for each \( t \) a bag.

The width of a decomposition is defined as \( \max(\{ |\beta(t)| - 1 \mid t \in V(\mathcal{T}) \}) \), then the treewidth of a graph \( G \) (denoted \( tw(G) \)) is the minimum width of a decomposition of \( G \) over all possible decompositions. A class of graphs then has bounded treewidth if there exists some constant \( k \) (or parameter in the context of Parameterized Complexity) such that every graph in the class has treewidth at most \( k \).

Let \( N^r_v \) denote the \( r \)-neighborhood of a vertex \( v \) in the graph \( G \). Given a graph \( G \) and an integer \( r \) the local treewidth (denoted \( ltw(G,r) \)) is \( \max(\{ tw(N^r_v) \mid v \in V \}) \). A class of graphs has bounded local treewidth if there exists some function \( f \) such that for every \( r \) every graph in the class has local treewidth at most \( f(r) \). A class has effectively bounded local treewidth if \( f \) is also computable. Graph classes that have bounded local treewidth (but not necessarily bounded treewidth) include graphs of bounded degree and planar graphs.

Given a graph \( G \) and \( d \in \mathbb{N} \), we say the graph \( H \) is a shallow minor (at depth \( d \)) if there exists a subgraph \( X \) of \( G \) such that the vertices of \( X \) can be partitioned into \( |H| \) disjoint, connected components of radius at most \( d \) such that contracting each component to a separate vertex gives the graph \( H \). Let \( G^d \) denote the set of shallow minors at depth \( d \) of \( G \). Given a class of graphs \( C \), let \( C^d \) denote \( \bigcup_{G \in C} G^d \). For any class of graphs \( C \), if there exists a function \( f \) such that \( sup_{C \in C^d} \frac{|C|}{|G|} \leq f(d) \), then we say \( C \) has bounded expansion. A graph class \( C \) is somewhere dense if there exists a \( d \) such that \( C^d \) contains every finite, simple graph. If no such \( d \) exists, \( C \) is nowhere dense. In particular, if \( C \) is nowhere dense, there exists a graph \( G \) (in fact, infinitely many such graphs) such that \( G \notin C^d \) for any \( d \). Bounded expansion and nowhere dense graphs were introduced by Nešetřil and Ossona de Mendez [31], and with Wood [32] they give a variety of characterizations of such classes.

Both treewidth and local treewidth can be transferred to general relational structures by taking the (local) treewidth of the Gaifman graph (q.v. Section 2.4).

2.3. Parameterized Complexity

Parameterized Complexity augments combinatorial problems using parameters as independent measures of structure in addition to the overall size of the input. An instance \((I,k)\) of a parameterized problem consists of the input \( I \), corresponding to the input of a classical problem and an integer parameter \( k \), a special part of the input independent from \( |I| \).

A problem is fixed-parameter tractable (or FPT) if there is an algorithm that solves each instance \((I,k)\) of the problem in time bounded by \( f(k) \cdot |I|^{|k|} \) where \( f \) is a computable function dependent only on \( k \).
Conversely hardness for any class in the \( W \)-hierarchy provides evidence that a problem is not fixed-parameter tractable. Hardness for such classes is typically established by parameterized reduction, the Parameterized Complexity reduction scheme where given an instance \((I, k)\) of problem \(\Pi_1\) an instance \((I', k')\) of problem \(\Pi_2\) is computed in time bounded by \(f(k) \cdot |I'|^{O(1)}\), with \(k' \leq g(k)\) for some computable function \(g\) and \((I, k)\) is a Yes-instance if and only if \((I', k')\) is a Yes-instance.

For more detail on the theory of Parameterized Complexity we refer the reader to [21], [33] and [15].

2.4. Logic, relational structures and the model checking problem

For a more detailed treatment of logic and finite model theory see the monographs of Ebbinghaus and Flum [17] and Ebbinghaus, Flum and Thomas [18].

A relational structure \(\mathcal{A}\) consists of a set \(A\) of elements called the universe and a set \(\tau\) of relations of finite arity called the vocabulary, along with an interpretation of each relation in \(\tau\) over \(A\). That is, for each relation \(R \in \tau\) of arity \(r\) there is a set of tuples from \(A^r\) that define when \(R\) is true. It is sufficient for our purposes to consider only structures where \(A\) is finite and \(\tau\) is finite and non-empty.

First order (FO) logic consists of a countably infinite set of (individual) variables and a vocabulary \(\tau\), which we will assume matches the vocabulary of the structure the logic is being applied to. Formulæ of first order logic are constructed from the variables, vocabulary, the atomic relation \(=\), the boolean connectives \(\lor, \land\) and \(\neg\) and the quantifiers \(\forall\) and \(\exists\). A formula is a sentence if there are no free variables, that is, all variables are bound by a quantifier.

Second order (SO) logic extends first order logic by including a countably infinite set of relation variables, but construction of formulæ is otherwise the same. If all relation variables are unary, then the logic is monadic (MSO). Note that the key difference between first and second order logics is not particularly that there are relation variables, but that second order logic allows quantification over them.

The precise semantics of applying a logic to a structure are unnecessary to discuss in detail here, suffice to say that (relation) variables of the formula are mapped to (sets of) elements of the universe and the formula is evaluated via the satisfaction of the model under the vocabulary of the structure.

Given a structure \(\mathcal{A}\) and a logical formula \(\phi\) the basic Model Checking problem is simply checking whether the formula \(\phi\) holds on \(\mathcal{A}\):

\[
\text{Model Checking (MC)}
\]

\[
\text{Instance: A structure } \mathcal{A}, \text{ and a formula } \phi.
\]

\[
\text{Question: } \mathcal{A} \models \phi?
\]

By \(\text{MC}(C, \Phi)\) we denote the model checking problem restricted to structures in the class \(C\) and formulæ in the class \(\Phi\).

There are three particularly interesting cases for the Model Checking problem, summarized by the following theorems.

**Theorem 1.** (See Courcelle [13–12].) Let \(C\) be any class of structures of bounded treewidth. \(\text{MC}(C, \text{MSO})\) is fixed-parameter tractable when parameterized by the treewidth of the structure and the length of the MSO sentence.

**Theorem 2** (Frick and Grohe [23]). Let \(C\) be any class of structures of bounded local treewidth. \(\text{MC}(C, \text{FO})\) is fixed-parameter tractable when parameterized by the length of the FO sentence.

**Theorem 3.** (See Dvořák, Král and Thomas [16].) Let \(C\) be any class of structures with bounded expansion. \(\text{MC}(C, \text{FO})\) is fixed-parameter tractable when parameterized by the length of the FO sentence.

Note that in all cases the length of the formula (and the treewidth) may themselves be dependent on other parameters.

A key tool in actually computing the solution to the model checking problem for many classes of structures is the Gaifman graph of the structure. Given a structure \(\mathcal{A}\), the Gaifman graph \(G^\mathcal{A}\) is the graph obtained by taking the universe \(A\) of \(\mathcal{A}\) as the vertex set of \(G^\mathcal{A}\), with an edge between two vertices if they ever appear in the interpretation of any relation in \(\tau\) together. Taking a graph as a structure with binary relation \(E\) (the edge relation), the Gaifman graph is precisely the graph.

3. Dynamic graphs as logical structures

Let \(G\) be a dynamic graph. We give two translations of \(G\) into a logical structure, with different properties and limitations appropriate to different applications.

3.1. Local Treewidth Preserving Structure

Let \(G = (V, E, T, \xi_v, \xi_e)\) be a dynamic graph.

Let \(\mathcal{A}(G)\) be the structure obtained from \(G\) with universe \(A\) and vocabulary \(\tau\) where \(A\) consists of
• an element $v^i$ for each $v \in V(G)$ if $t \in \zeta_v(v)$,
• an element for each $t \in T(G)$ for all $t \in T$.

equipped with a function $f_{uv}$ such that $f_{uv}(v^i) = f_{uv}(u^i)$ if and only if $v^i$ and $u^i$ are derived from the same vertex $v \in V(G)$.

The vocabulary $\tau = \{V^A, E^A, T^A, R^A, L_t^A\}$ where

• $v \in V^A(G)$ if and only if $v \in V(G)$,
• $uv \in E^A(G)$ if and only if $uv \in E(G)$,
• $t \in T^A(G)$ if and only if $t \in T(G)$,
• $(t_1, t_2) \in R^A(G)$ if and only if $t_1, t_2 \in T(G)$ and $t_1 \leq t_2$,
• $v \in L_t^A(G)$ if and only if $t \in \zeta_v(v)$, and
• $uv \in L^A(G)$ if and only if $t \in \zeta_v(uv)$.

When the structure is understood from context, we will drop the $A$ superscript.

**Theorem 4.** Given a dynamic graph $G$, if $G_t$ has local treewidth (effectively) bounded by $f(r)$ at each time $t \in T(G)$, then the Gaifman graph $G$ obtained from $A(G)$ has local treewidth (effectively) bounded by $\max[f(r), |T(G)| - 1]$.

**Proof.** Let $G_t$ be the graph at time $t \in T(G)$ and $G_t$ the corresponding component of $G$, $G_t$ is essentially $G_t$.

As $G_t$ has (effectively) bounded local treewidth, there is a function $f$ such that for each $v \in V$ and $r \in \mathbb{N}$ we have $\text{tw}(|N_t(v)|) \leq f(r)$ in $G_t$ (note that this may not be true if we consider $G$ in total, ignoring the timing of the edges). In particular for each $v$ and $r$ in each $G_t$, we have a tree decomposition where each bag has at most $f(r) + 1$ elements. Moreover we have such a decomposition at each time $t$, with no interaction between any two snapshots. Therefore the width of the decomposition is at most $f(r)$.

The only component of $G$ that does not have an appropriate decomposition is the clique defined by the time elements representing $T(G)$. The only possibly decomposition is to place all elements in one bag, with width $|T(G)| - 1$.

As each $G_t$ and the clique are disjoint, there are no additional edges to consider in the possible decompositions. \(\square\)

We note also that if the graph has bounded degree at every $t \in T(G)$, this property is also preserved in the Gaifman graph. With the construction of Section 3.3, planarity can also be preserved.

Although preservation of local treewidth is of particular interest for practical reasons discussed in Section 5, this logical structure in fact preserves a more general property: bounded expansion.

**Theorem 5.** Let $\mathcal{C}$ be a graph class with expansion bounded by the expansion function $f$. Let $\mathcal{D}$ be the family of dynamic graphs $G$ such that $G_t \in \mathcal{C}$ for all $t \in T$. Let $\mathcal{G}$ be the family of Gaifman graphs derived from the structures $A(G)$ where $G \in \mathcal{D}$. The class $\mathcal{G}$ has bounded expansion.

**Proof.** As $\mathcal{C}$ has expansion bounded by the function $f$, we know that for each graph $G \in \mathcal{D}$, for each $t \in T$:

$$\sup_{H \in G_t \subset V} \frac{\|H\|}{|H|} \leq f(d)$$

The Gaifman graph $G$ of the structure $A(G)$ for any graph $G \in \mathcal{D}$ consists of $t + 1$ connected components. Each $G_t$ contributes a single connected component to the graph itself, the final component is the graph defined by $T(G)$ under the relation $R$. As the set $H \subset V$ for any $H$ and $d$ is constructed by the basic action of contracting connected subgraphs of radius at most $d$, each graph in $G \subset V$ for each $d$ is constructed from the disjoint union $\biguplus_{\mathcal{G}} G_t \subset V$ of graphs from $G_t \subset V$ for each $t$ and $T \subset V$ where $T$ is the time component of $G$.

For each $G \in \mathcal{G}$ we therefore have

$$\sup_{G' \in G' \subset V} \frac{\|G\|}{|G|} = \sup_{H \in H \subset G' \subset V} \frac{\|H\|}{|H|}$$

As $H$ is composed of disjoint components, there is some graph $H_t \in G_t \subset V$ for each time $t$ and some graph $T \in T \subset V$ such that

$$\sup_{H \in H \subset G_t \subset V} \frac{\|H\|}{|H|} = \frac{\sum_{t} |H_t| + |T|}{\sum_{t} |H_t| + |T|}$$

This is in turn at most

$$\sum_{t} \frac{|H_t|}{|H_t|} + \frac{|T|}{|T|} \leq |T| \cdot f(d) + \frac{|T| - 1}{2}$$
Thus for each fixed $|T|$ there exists a function $f'$ such that for all $d$ we have
\[
\nabla_d \mathcal{G} = \sup_{H \in \mathcal{G}^d} \frac{\|H\|}{|H|} \leq f'(d)
\]
so $\mathcal{G}$ has bounded expansion.

Although not algorithmically interesting, we note that this structure also preserves the property of being nowhere dense, which includes graphs of bounded expansion.

**Lemma 6.** Let $\mathcal{C}$ be a nowhere dense graph class. Let $\mathcal{D}$ be the family of dynamic graphs $G$ such that $G_t \in \mathcal{C}$ for all $t \in T$. Let $\mathcal{G}$ be the family of Gaifman graphs derived from the structures $\mathcal{A}(G)$ where $G \in \mathcal{D}$. The class $\mathcal{G}$ is nowhere dense.

**Proof.** As $\mathcal{C}$ is nowhere dense, there exists a graph $X$ such that $X \not\in \mathcal{C} \forall d$ for every $d$. In particular it is sufficient to assume that there exists a $k$ such that $X = K_k$ (let $X$ be any graph that witnesses that $\mathcal{C}$ is nowhere dense, then we immediately have that $K_{|X|} \not\in \mathcal{C} \forall d$ for any $d$, otherwise there exists a $d$ such that $X \in \mathcal{C} \forall d$, as $X$ can be obtained from $K_{|X|}$).

Moreover any $K_k$ with $k' > k$ is also excluded from $\mathcal{C} \forall d$ for all $d$. As every $G \in \mathcal{G}$ is a disjoint union of graphs from $\mathcal{C}$ with an additional connected component for the time elements which we may assume is isomorphic to $K_{|T|}$, then there exists some $l \geq |T| + 1$ such that $K_l \not\in \mathcal{C} \forall d$ for any $d$. □

### 3.2 Treewidth Preserving Structure

If we are only concerned with the treewidth of the structure, we can use a somewhat more natural structure.

Let $G = (V, E, T, \tau, \zeta, \xi)$ be a dynamic graph.

In this case the universe $A$ consists of an element $v^t$ for each $v \in V$ if $t \in \zeta_v(v)$, an element for each $t \in T(G)$, and $\tau = \{V^A, E^A, S^A, T^A, R^A\}$ where $v^t \in V^A(G)$ if and only if $v \in V(G)$ and $t \in \zeta_v(v)$, $v^t \in E^A(G)$ if and only if $v^t \in E(G)$ and $t \in \zeta(v)$, $v^t \in T^A(G)$ if and only if $t \in T(G)$, and $(t_1, t_2) \in R^A(G)$ if and only if $t_1, t_2 \in T(G)$ and $t_1 < t_2$.

**Theorem 7.** If $G$ has $tw(G) \leq k$ at every time $t \in T(G)$ then the Gaifman graph $G^t$ obtained from $\mathcal{A}(G)$ has $tw(G^t) \leq \max[k + 1, |T(G)| - 1]$.

**Proof.** Let $G_t$ be the snapshot of $G$ at time $t \in T(G)$ and let $tw(G_t) \leq k$ for every $t \in T(G)$. The Gaifman graph $G_t$ consists of $|T(G)| + 1$ components; a component $G_{t'}$ for each $G_t$ which is the normal Gaifman graph translation of a graph, i.e. the graph itself, and a $|T(G)|$-clique corresponding to the time elements of $A$. Each element of the time clique is connected to every vertex in exactly one $G_{t'}$ (precisely the element corresponding to time $t \in T(G)$) and each vertex in each $G_{t'}$ is connected to exactly one vertex of the time clique.

By assumption for each $G_t$, we have $tw(G_t) \leq k$. Thus each $G_t$ has a tree decomposition where each bag has at most $k + 1$ elements. To each of these bags we add the element corresponding to time $t \in T(G)$. The time clique forms a bag of size $|T(G)|$. We add an edge to the tree decomposition between the time clique bag and an arbitrarily chosen bag from each decomposition of each $G_t$. Thus we have a tree decomposition of $G$ with bag size at most $\max[k + 2, |T(G)|]$ and therefore $tw(G) \leq \max[k + 1, |T(G)| - 1]$. □

### 3.3 Structures for totally ordered time

If we can assume that time is linear for our dynamic graph $G$, i.e. there exists a total order on the elements of $T(G)$ (hence for every pair $t_i, t_j \in T(G)$ either $t_i < t_j$ or $t_j < t_i$), we may construct a structure where the Gaifman graph avoids the clique created by the elements corresponding to the times. The trade-off is that the construction of logical sentences becomes more involved in the sense that the sentences become longer.

To obtain this modified structure we restrict the relation $R$ in each $\tau$ such that $(t_i, t_j) \in R$ if and only if $t_1$ immediately precedes $t_2$. Furthermore we add a constant $s$ such that $(s, t_i) \in R$ if and only if $t_i$ is the earliest time element. Then the component of the Gaifman graph constructed from $\mathcal{A}(G)$ corresponding to $R$ is a path, where each $t_i$ has an associated vertex, and there is an edge between two such vertices if and only if one immediately precedes the other temporally. Moreover we can define the sentence between each time element and $s$ recursively: $d_1(t_i) := R_{t_i}^t$ and $d_{n+1}(t_i) := \exists \ell(T_{t_j} \land R_{t_j}^{t_i} \land d_{n-1}(t_j))$.

Note of course that as first order logic is compact, these formulae must be defined separately for each $|T(G)|$, giving a finite set in each case. As $|T(G)|$ will always be taken as a parameter we effectively produce an infinite family of such formulae, and select the appropriate subset for each instance of whichever problem we deal with. In the case of second order logic this problem does not exist, but it is convenient to use the first order construction. Then within the logic we can define the order relation over the time elements as $t_i \leq t_j := (t_i = t_j) \lor \bigvee_{i \in [1, |T(G)|]} (d_i(t_i) \land \neg d_i(t_j))$. 
The operator $\preceq$ allows comparison over all times, but at the cost of formulae which may depend on the size of $T(G)$. However there is a notable change in treewidth and local treewidth of the structures using this approach.

**Theorem 8.** Given a dynamic graph $G$, let $\mathcal{A}(G)$ be the structure constructed from $G$ as a Local Treewidth Preserving Structure, except with a linear time relation $R$.

If $G$ has local treewidth (effectively) bounded by $f(r)$ at every $t \in T(G)$, then the Gaifman graph $\mathcal{G}$ derived from $\mathcal{A}(G)$ has local treewidth (effectively) bounded by $f(r)$.

**Proof.** The only change to the structure in this case is the component concerned with the time elements. In the first construction this was a clique of size $|T(G)|$, but now is a path of length $|T(G)|$. As trees have treewidth 1, and local treewidth is bounded by treewidth, the local treewidth of this component is at most 1. □

**Theorem 9.** Given a dynamic graph $G$, let $\mathcal{A}(G)$ be the structure constructed from $G$ as Treewidth Preserving Structure, except with a linear time relation $R$.

If $G$ has $\text{tw}(G) \leq k$ at every $t \in T(G)$, then the Gaifman graph $\mathcal{G}$ derived from $\mathcal{A}(G)$ has $\text{tw}(\mathcal{G}) \leq k + 1$.

**Proof.** As before each component corresponding to the graph $G_t$ at time $t$ has treewidth at most $k$, and as before we add $t$ to each bag of this decomposition. For the time component we construct the following tree decomposition:

1. each vertex $t$ is given a bag labeled $\{t\}$,
2. each edge $st$ is given a bag labeled $\{s,t\}$ and
3. each vertex bag is connected in the decomposition to the two edge bags that contain the same label.

This component clearly has treewidth 1. Then we complete the decomposition for $\mathcal{G}$ by adding a decomposition edge from the vertex bag for time element $t$ to an arbitrary bag in the decomposition of $G_t$. Clearly this is still a tree, and the whole decomposition has width at most $k + 1$. □

Note that for both representations of the temporal element of the graph, the complexity of the problem is at least partially dependent on $|T(G)|$, in that either the (local) treewidth depends on $|T(G)|$, or the length of any formula where we have to compare times does.

In the bounded expansion case, we now have that $\frac{1 + k}{1 + k} < 1$, so the function $f'$ remains a suitable expansion function.

**4. Applications to dynamic graph problems**

**4.1. Adapting the metatheorems to the dynamic context**

We can now simply extend the metatheorems appropriately:

**Theorem 10.** Let $G$ be a dynamic graph with $\text{tw}(G) \leq k$ at every time $t$ and $\phi$ a sentence of monadic second order logic. The problem $MC(G, \phi)$ is fixed-parameter tractable with parameter $k + |\phi| + |T(G)|$.

**Proof.** Courcelle’s Theorem gives us that the model checking problem for graphs of bounded treewidth is fixed-parameter tractable with the treewidth and the length of the formula as a combined parameter.

In our case we may apply Courcelle’s Theorem to the Gaifman graph derived from the dynamic graph. As the treewidth of the Gaifman graph is bounded by the treewidth of the original graph and $|T(G)|$, we obtain our result. Note that in the case where we use the linear time structure variant, $|T(G)|$ is implicitly included in $|\phi|$, if necessary. □

**Theorem 11.** Let $G$ be a dynamic graph with effectively bounded local treewidth at every time $t$ and $\phi$ a sentence of first order logic. The problem $MC(G, \phi)$ is fixed-parameter tractable with parameter $|\phi| + |T(G)|$.

**Proof.** By Frick and Grohe’s Theorem, the model checking problem for graphs of effectively bounded local treewidth is fixed parameter tractable with the length of the formula as the parameter. Again the length of the formula may implicitly depend upon other parameters as part of the problem.

Stewart notes that Frick and Grohe’s Theorem still holds if the bound on the local treewidth depends upon a parameter, rather than simply being constant. In the case where we do not use the linear time construction, the local treewidth is bounded by $\max\{f(r), |T(G)| - 1\}$. □

**Theorem 12.** Let $\mathcal{C}$ be a graph class with bounded expansion, $G$ a dynamic graph such that for every time $t$ the snapshot graph $G_t \in \mathcal{C}$ and $\phi$ a sentence of first order logic. The problem $MC(G, \phi)$ is fixed-parameter tractable with parameter $|\phi| + |T(G)|$. 
**Proof.** By Theorem 5, the class $\mathfrak{G}$ formed of Gaifman graphs of structures constructed from such graphs $G$ also has bounded expansion. The result then follows from Dvořák, Král and Thomas’ Theorem.

We demonstrate the use of these theorems by application to some dynamic graph problems of general interest. Bhadra and Ferreira [2] prove that the problem of finding a connected component of size at least $k$ in an evolving digraph is $NP$-complete.

**Strongly Connected Dynamic Component (SCDC)**

**Instance:** A dynamic digraph $D = (V, E, T, \zeta_v, \zeta_e)$, an integer $k$.

**Parameter:** $k + |T|$.

**Question:** Is there a set $V' \subseteq V$ with $|V'| \geq k$ such that for every pair $u, v \in V'$ we have $J(u, v)$?

Adapting the structures for directed graphs is simply a matter of semantics for the $E$ relation in $\tau$. Using this we apply the metatheorem to demonstrate that the problem is fixed-parameter tractable.

**Lemma 13.** SCDC is expressible in first order logic.

**Proof.** We first define a sentence that expresses the idea of a journey:

$\mathcal{J}_n(u, v) := \mathcal{J}_{n-1}(u, v) \lor \exists x_1, \ldots, x_{n+1}, t_1, \ldots, t_n \left( \bigwedge_{i \in [1,n+1]} Vx_i \land \bigwedge_{i \in [1,n]} \neg Tt_i \right) \land \left( \bigwedge_{i \in [1,n]} Ex_i x_{i+1} \land L_t x_i x_{i+1} \right) \land \left( \bigwedge_{i \in [1,n]} t_i \leq t_{i+1} \right) \land \left( f_V(u) = f_V(x_1) \land f_V(v) = f_V(x_{n+1}) \right)$

Using this the problem can be succinctly defined in first order logic as: $\exists v_1, \ldots, v_k \forall x, y \left( f_V(x) = f_V(v_i) \land f_V(y) = f_V(v_j) \Rightarrow \mathcal{J}_k(x, y) \right)$. Note that the length of the sentence is a function of $k$ and $|T(G)|$ and that we again avoid running afoul of the compactness of first order logic by restricting the possible journey lengths to $k$ for each instance.

Combining Theorem 11 and Lemma 13 we obtain the tractability result.

**Corollary 14.** SCDC is fixed-parameter tractable for dynamic graphs of bounded local treewidth.

As first order logic is a subset of monadic second order logic we may also use Theorem 10.

**Corollary 15.** SCDC is fixed-parameter tractable for dynamic graphs of bounded treewidth.

Of course using Theorem 12 gives:

**Corollary 16.** SCDC is fixed-parameter tractable for dynamic graphs of bounded expansion.

Another interesting dynamic graph problem is whether a sending vertex can receive a reply without the reply having to travel too far.

**Short Message Return Path (SMRP)**

**Instance:** A dynamic digraph $D = (V, E, T, \zeta_v, \zeta_e)$ with an identified vertex $v \in V$ and an integer $k$.

**Parameter:** $k + |T|$.

**Question:** Is there a journey from each $u \in N^{out}(v)$ to $v$ of length at most $k$?

**Lemma 17.** SMRP is expressible in first order logic.

**Proof.** Using the ability to express the idea of a journey in first order logic, SMRP is simple to express: $\forall u \left( Vu \land Evu \Rightarrow \exists u'(f_V(u) = f_V(u) \land \mathcal{J}_k(u, v)) \right)$. Using Theorems 11, 10 and 12 with Lemma 17 we obtain the following tractability result.
Corollary 18. Short Message Return Path is fixed-parameter tractable for dynamic graphs of bounded local treewidth, bounded treewidth and bounded expansion.

4.2. Transferring static results

The transference of the metatheorems also implies the transference of previous results on static graphs, with only minor changes in problem formulation. Most immediately, these results hold where we relax requirements such that the edges required for the solution exist at some point.

We demonstrate by “temporalizing” k-Coloring and the associated MSO formula that demonstrates its fixed-parameter tractability for graphs of bounded treewidth.

**Permanent Coloring**

**Instance:** A dynamic graph \( G = (V, E, T, \zeta_v, \zeta_e) \), an integer \( k \).

**Parameter:** \( k + |T| \).

**Question:** Is there a static assignment of \( k \) colors to \( V \) that is a proper coloring of \( G \) at each time \( t \in T \)?

**Lemma 19.** **Permanent Coloring** is expressible in monadic second order logic.

**Proof (Sketch).**

\[
\exists V_1, \ldots, V_k \forall x, y, t \left( \bigvee_{i \in [1, k]} \left( V_i x \land \lnot V_j x \right) \land \bigvee_{i \in [1, k]} \left( V_i y \land \lnot V_j y \right) \land V x \land V y \land T t \land \text{Exy} \right) \\
\rightarrow \bigwedge_{i \in [1, k]} \lnot (V_i x \land V_i y) \quad \square
\]

Alternatively we can reframe the problem as requiring a solution that is true at every point in time.

**Evolving Coloring**

**Instance:** A dynamic graph \( G = (V, E, T, \zeta_v, \zeta_e) \), an integer \( k \).

**Parameter:** \( k + |T| \).

**Question:** Is there a (possibly different) proper \( k \)-coloring of \( G \) at each time \( t \in T \)?

**Lemma 20.** **Evolving Coloring** is expressible in monadic second order logic.

**Proof (Sketch).**

\[
\forall t \exists V_1, \ldots, V_k \forall x, y \left( \bigvee_{i \in [1, k]} \left( V_i x \land \lnot V_j x \right) \land \bigvee_{i \in [1, k]} \left( V_i y \land \lnot V_j y \right) \land V x \land V y \land T t \land \text{Exy} \right) \\
\rightarrow \bigwedge_{i \in [1, k]} \lnot (V_i x \land V_i y) \quad \square
\]

Then by Theorem 10:

**Corollary 21.** **Permanent Coloring** and **Evolving Coloring** are fixed parameter tractable on graphs of bounded treewidth.

While illustrative, these results for dynamic coloring problems are not particularly distinct from the static version; computing an evolving coloring is polynomially equivalent to computing a coloring at each point in time, and computing a permanent coloring is equivalent to computing a coloring of the graph that is the union of the graphs at each point in time.

For other problems, the relationship between static and dynamic problems is not as simple.
Whereas a permanent coloring must also be a coloring of the union of the snapshot graphs, a minimum permanent dominating set may be smaller than a minimum dominating set of the union of the snapshot graphs, but is obviously lower bounded by the size of the minimum dominating set of any of the snapshot graphs.

**Lemma 22.** **Permanent Dominating Set** is expressible in first order logic.

**Proof (Sketch).**

\[ \exists v_1, \ldots, v_k \forall t \forall u \left( \left( V_u \land T_t \land L_t u \right) \rightarrow \exists v \left( V_v \land L_t v \land \left( \bigvee_i v = v_i \right) \land L_t u v \right) \right) \tag{2} \]

By Theorem 12:

**Corollary 23.** **Permanent Dominating Set** is fixed-parameter tractable on graphs of bounded expansion.

It is easy to see that, although the formulæ become somewhat more complex, it is a simple matter to translate existing expressibility results (often already in the context of parameterized tractability) into results for dynamic graphs. Thus the permanent and evolving versions of interesting problems such as **Independent Set**, **Vertex Cover**, **Clique** and **Subgraph Isomorphism** are fixed-parameter tractable on dynamic graphs of bounded (local) treewidth (Flum and Grohe [21] give appropriate static results).

5. Dynamic graph classes with bounded (local) treewidth

One broad class of dynamic graphs that has proven interesting due to the connectivity problems induced are sparse dynamic graphs. There are several definitions of what constitutes a sparse graph in the dynamic context, with two common definitions involving Markovian processes and bounded degree graphs. Both of these classes in fact have bounded local treewidth.

Flum and Grohe [21] note that given a graph \( G \) of maximum degree \( d \) we have \( \text{ltw}(G, r) \leq d' \). Thus the class of bounded degree dynamic graphs has bounded local treewidth.

As mentioned, sparse graphs may also be defined by Markovian processes on the edges of the graph where the probability of an edge existing, and continuing to exist are bounded (e.g., [10]). This also leads to bounded local treewidth for such graphs.

**Theorem 24.** Let \( G = (G_1, \ldots, G_t) \) be an edge-Markovian dynamic graph where the probability of a non-edge becoming an edge is \( p \leq c_1/n \) and the probability of an edge becoming a non-edge is \( q \leq 1 - c_2/n \), then the expected maximum degree for each \( G_{i \geq 2} \) is at most \( c_1 + c_2 \).

**Proof.** At any time \( t \), for each vertex \( v \in V(G) \) we have

\[
|N_{t+1}(v)| = (1 - q) \cdot |N_t(v)| + p(n - 1 - |N_t(v)|) \\
\leq \frac{c_2}{n} \cdot |N_t(v)| + \frac{c_1}{n} \cdot n - \frac{c_1}{n} \cdot |N_t(v)| - \frac{c_1}{n} \\
= \frac{c_2}{n} \cdot \frac{|N_t(v)|}{n} + c_1 \cdot \frac{|N_t(v)| + 1}{n} \\
\leq c_1 + c_2 \tag{3}
\]

**Corollary 25.** Let \( G = (G_1, \ldots, G_t) \) be an edge-Markovian dynamic where the probability of a non-edge becoming an edge is \( p \leq c_1/n \) and the probability of an edge becoming a non-edge is \( q \leq 1 - c_2/n \), then \( G \) is expected to have bounded local treewidth.

We note that the third common definition for sparse dynamic graphs, using some measure of density of the graph, does not lead to bounded local treewidth. Some measures of density lead to bounded local treewidth, but essentially by implying bounded degree.

6. Conclusion

The structural parameters treewidth and local treewidth have proven to be very useful in algorithmic design when dealing with hard problems. Aiding in this task are the metatheorems of Courcelle and Frick and Grohe that provide complexity classifications for large classes of graphs using logical expressibility. As dynamic graph theory and dynamic graph problems become more prominent thanks to the advance of technology, questions of treewidth arise. We have shown that given a
finite, bounded temporal context the notions of logical expressibility and the corresponding metatheorems can be extended to provide tools for classification of problems on dynamic graphs. In particular if a dynamic graph has bounded (local) treewidth at every point in time, then we can construct a logical structure, and hence Gaifman graph where the (local) treewidth remains bounded. Furthermore in the finite, bounded temporal context first order and thus second order logic can be augmented (finitely) with relations that can express the temporal notion of before and after. With these tools we can easily classify many problems, including temporal extension of static problems.

A natural focus for extending this work would be to examine other measures of sparseness used with dynamic graphs to determine, despite not preserving treewidth or local treewidth, whether they satisfy the more general bounded expansion property.

This approach however is limited by the fact that we require the length of the time interval to be finite and a parameter. If the length of time is unbounded (even finite but unbounded), then we do not have the logical expressibility to deal with these problems. Thus certain classes of dynamic graphs are excluded by this condition. However we can still easily express properties over a finite amount of time, or in periodic graphs, which covers a significant amount of real applications [7,20], where finite time is a constraint of the setting, rather than an artificial stipulation of the theory. If we have a total ordering on the temporal domain and we do not need to compare time elements that are too far apart, then we can completely remove the dependence on the length of the time interval. It is an open question as to whether this dependence can be removed in general.

References