Symbolically Bounding the Drift in Time-Constrained MSC Graphs.

S. Akshay[1], Blaise Genest[2], Loïc Hélouët[3], and Shaofa Yang[4]

1 **National University of Singapore.**
2 **CNRS, UMI IPAL joint with NUS and A*STAR/I2R, Singapore.**
3 **INRIA Rennes, France.**
4 **SIAT, Chinese Academy of Sciences, China.**

Abstract

Systems involving both time and concurrency are notoriously difficult to analyze. Existing decidability results apply in settings where clocks on different processes cannot be compared or where the set of timed executions is regular. We prove new decidability results for timed concurrent systems, requiring neither restriction. We consider the formalism of time-constrained MSC graphs (TC-MSC graphs for short), introduced in [2]. We study the problem of checking whether the set of timed executions generated by a TC-MSC graph is empty, which is undecidable in general [9].

In this paper, we show the decidability of this problem under the restriction that every path of the TC-MSC graph is prohibited from *forcing* any basic scenario labeling a node to take more than K units of time to complete, for a given K. Further, we prove that this condition can be effectively checked. Our approach consists in encoding the time constraints seen along a path into a bounded system of inequalities. Instead of constructing an interleaved model and using zones of timed automata as in existing approaches, we symbolically manipulate the system of inequalities using the Fourier-Motzkin elimination method. This allows for decision procedures which are both efficient and handle non regular specifications.

1 Introduction

In a distributed system, several processes interact to implement a collection of global behaviors. Protocol specifications include timing requirements for messages as well as descriptions of how to recover from timeouts. Thus, a protocol designer has to deal with situations where time and concurrency influence each other. One way to describe these interactions is through scenarios, formalized using Message Sequence Charts (MSCs) [11]. The timing information is captured by adding timing constraints between pairs of events, yielding time-constrained MSCs (denoted TC-MSCs). A protocol is then described by allowing choices and repetition of scenarios. To specify these main characteristics of protocols while abstracting away details of implementation, the formal methods community often considers *MSC graphs*, which are directed graphs whose nodes are labeled by MSCs. MSC graphs have been generalized to *time-constrained MSC graphs* (TC-MSC graphs) [2], whose nodes are labeled by TC-MSCs and edges have additional timing constraints. In general, such models do not have regular sets of executions. In this paper, we consider decidability issues for TC-MSC graphs.

Obtaining decidability in the presence of both time and concurrency is a challenging issue. For instance, even checking whether there exists a timed execution that is consistent with all the constraints of a model is non trivial. This question, called the *emptiness problem*, is undecidable for TC-MSC graphs in general [9]. However, it is decidable for (sequential) timed automata [3]. Extending decidability results to distributed systems has been done only in two particular and limited settings. In the first setting, [13, 8] consider clocks that are local to a process. But then, one cannot specify time taken by a communication (message or synchronization). This limitation makes the specification formalism very weak. The second

setting can relate clocks from different processes and specify how long a communication can or must take [2, 1, 5, 6]. However, these papers restrict the concurrency in a structural way, for instance considering only locally synchronized (see [14, 4, 10]) MSC graphs (in [2, 1]) or only safe Petri Nets (in [5, 6]). The language of the specification is then forced to be regular, which is a significant restriction in a concurrent setting where even simple behaviors may not be regular (e.g., the producer-consumer protocol). Also, the procedures for TC-MSC graphs in [2, 1, 9] rely on the construction of an interleaved model, leading to a combinatorial explosion. This could be seen as a contradiction to the spirit of MSCs, which tries to avoid interleavings. Further, the approaches in [2, 1, 9] ultimately use zone construction techniques for timed automata, which implies another blow up in complexity.

In this paper, we propose the first decidability result for timed concurrent systems with global clocks having a possibly *non regular* set of behaviors. We investigate the emptiness problem for TC-MSC graphs, and prove it to be decidable in the setting where a TC-MSC graph is prohibited from *forcing* any basic scenario to take an arbitrarily long amount of time to complete. More precisely, for some given integer K, for any path ρ of a TC-MSC graph, if there exists at least one execution of ρ, then we require that there exists at least one in which the occurrence times of any two events from the same basic scenario differ by at most K. Such a TC-MSC graph is said to be K-drift-bounded. We further show that given K, one can effectively test whether a TC-MSC graph G is K-drift-bounded. Both results are established without constructing an interleaved model nor zones obtained from a timed automaton and thus avoids both state space explosions. Instead, we translate time constraints of any path of a TC-MSC graph into a symbolic profile, in the form of a system of inequalities. We show how to manipulate this system symbolically using Fourier-Motzkin elimination and Shostak lemma [15]. We then show that symbolic profiles can be approximated by a bounded system of inequalities whose coefficients are integers in $[-K, K]$, which is safe for K-drift-bounded paths. This forms the cornerstone of our decidability results, as these systems of inequalities can be recognized by a finite state automaton.

The paper is organized as follows: Section 2 recalls basic definitions of TC-MSC graphs. Section 3 defines drift-boundedness and discuss its relevance. Section 4 shows how to check emptiness for K-drift-bounded TC-MSC graphsand Section 5 shows that K-drift-boundedness is decidable, for a given K.

2 Preliminaries

Through the rest of the paper, we fix a finite set $\mathcal{P}$ of processes and let p, q range over $\mathcal{P}$. Let $\Sigma = \{p!q, p?q \mid p, q \in \mathcal{P},\ p \neq q\}$ be the *communication alphabet*. The letter $p!q$ represents p sending a message to q, while $p?q$ signifies p receiving a message sent by q. We define the map $loc : \Sigma \to \mathcal{P}$ via $loc(p!q) = p = loc(p?q)$, and call $loc(a)$ the *location* of a.

We now give the definition of Message Sequence Charts (MSCs) and time-constrained MSCs (TC-MSCs). In this paper we do not need the FIFO assumption on the messages.

▶ **Definition 1.** An *MSC* is a tuple $(E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda)$ where E is a finite set of events and $\lambda : E \to \Sigma$ is a labeling function. For each p, $<_p$ is a total order over events of $E_p = \{e \in E \mid loc(\lambda(e)) = p\}$. The message function $\mu \subseteq E_S \times E_R$ is a bijection, such that $f = \mu(e)$ implies $\lambda(e) = p!q$, $\lambda(f) = q?p$ for some $p, q \in \mathcal{P}$, with $E_S = \{e \in E \mid \lambda(e) = p!q, \text{ for some } p, q \in \mathcal{P}\}$ and $E_R = \{f \in E \mid \lambda(f) = q?p, \text{ for some } p, q \in \mathcal{P}\}$. We require that $\lessdot = \bigcup_{p\in\mathcal{P}} <_p \cup \mu$ is such that its transitive closure $\leq$ is a partial order.

The relation $\leq$ reflects causal ordering of events. We will write $e < f$ when $e \leq f$ and

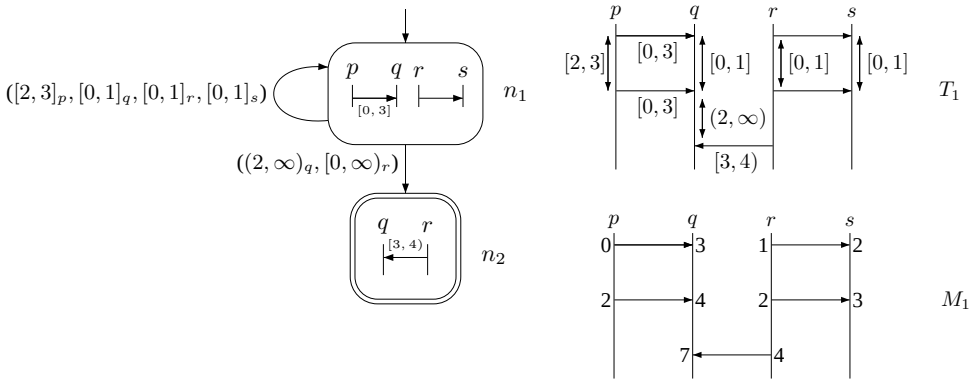


Figure 1 A TC-MSC graph G_1, a TC-MSC T_1 induced by the path $n_1 \cdot n_1 \cdot n_2$ of G_1 and a dated MSC $M_1 \in \mathcal{L}(G_1)$ generated by T_1

$e \neq f$. Notice that E_p has a unique $<_p$-maximal event (respectively, minimal event), which we refer to as the last (respectively, first) event of E on p.

Let $\mathbb{N}$ be the set of natural numbers. Let $\mathcal{I}$ denote the collection of intervals of open and closed intervals with end points in $\mathbb{N}$ as well as intervals of the form $[c, \infty)$ where $c \in \mathbb{N}$.

▶ **Definition 2.** A *TC-MSC* is a structure $(E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda, \delta)$ where $(E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda)$ is an MSC and δ is a function which associates an interval $\delta(e, e') \in \mathcal{I}$ to each $e \lessdot e'$.

For each pair of events $e \lessdot e'$, the interval $\delta(e, e')$ constrains the range in which the difference between the occurence time of e' and the occurrence time of e can lie. For clarity, we shall refer to occurrence times as *dates*. Though we have considered only constraints as intervals with natural end-points, our results extend easily to constraints expressed as intervals involving rational end-points. A TCMSC T defines a collection of scenarios with dates such that relative difference of dates fulfill the constraints asserted in T.

▶ **Definition 3.** Let $(E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda, \delta)$ be a TC-MSC. A *dated MSC generated by* T is a structure $(E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda, d)$ where d is a function which associates a non-negative real to each event in E and such that for each $e \lessdot e'$, $d(e') - d(e)$ is in the interval $\delta(e, e')$.

We denote by $\mathcal{L}(T)$ the set of dated MSCs generated by T. To capture possibly infinite collections of TC-MSCs, we define TC-MSC graphs, which are finite graphs whose nodes are labeled by TC-MSCs. Each path ρ of a TC-MSC graph G induces a TC-MSC by concatenating TC-MSCs labeling nodes of ρ. Transitions of G are labeled with interval constraints, one for each process, that act as constraints on the timing between the last and the first event of consecutive nodes in ρ.

▶ **Definition 4.** A TC-MSC graph is a structure $G = (N, \mathcal{T}, \Lambda, n_{in}, N_{fi}, \longrightarrow, \Delta)$ where N is a finite nonempty set of nodes, $\mathcal{T}$ a finite set of TC-MSCs, $\Lambda : N \to \mathcal{T}$ labels each node with a TC-MSC, n_{in} is the initial nodes, N_{fi} the set of final nodes, $\longrightarrow \subseteq N \times N$ is the transition relation, and Δ is a labelling function which associates an interval $\Delta_p(n{\to}n') \in \mathcal{I}$ to each transition $n{\to}n'$ and each process p. We call a TC-MSC graph *full* if each node has events on every process in $\mathcal{P}$.

A path ρ of the TC-MSC graph G is a sequence $n_0 n_1 \dots n_\ell$ such that $n_0 = n_{in}$ and $n_i{\to}n_{i+1}$ for $i = 0, \dots, \ell - 1$. The path ρ is said to be *final* if $n_\ell \in N_{fi}$.

Let us now define the concatenation of TC-MSCs labeling adjacent nodes in G. For each $n{\to}n'$, the *concatenation* of TC-MSCs $\Lambda(n)$, $\Lambda(n')$ is defined with respect to $\Delta(n{\to}n')$,

and is denoted $\Lambda(n) \circ \Lambda(n')$. Roughly speaking, concatenation of $\Lambda(n)$ and $\Lambda(n')$ consists in putting $\Lambda(n')$ after $\Lambda(n)$ and for every process $p \in \mathcal{P}$, attaching to the pair (e_p, f_p) the constraints $\Delta_p(n \to n')$, for e_p the last event of $\Lambda(n)$ on process p and f_p the first event of $\Lambda(n')$ on process p.

Formally, let $\Lambda(n) = (E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda, \delta)$, $\Lambda(n') = (E', (<'_p)_{p\in\mathcal{P}}, \mu', \lambda', \delta')$. Then $T = (E'', (<''_p)_{p\in\mathcal{P}}, \mu'', \lambda'', \delta'')$ where E'' is the disjoint union of E and E', $<''_p$ is the transitive closure of the union of $<_p$, $<'_p$ and $E_p \times E'_p$, and λ'' is given by: $\lambda''(e) = \lambda(e)$ for $e \in E$, $\lambda''(e) = \lambda'(e)$ for $e \in E'$. We also set $\mu''(e) = \mu(e)$ when $\mu(e)$ is defined, and $\mu''(e) = \mu'(e)$ when $\mu'(e)$ is defined. At last, δ'' is given by: $\delta''(e, f) = \delta(e, f)$ for $e \lessdot f$, $\delta''(e, f) = \delta'(e, f)$ for $e \lessdot' f$. For each p, if both E_p and E'_p are nonempty, setting e_p the last event of E_p and f_p the first event of E'_p, $\delta''(e_p, f_p) = \Delta_p(n \to n')$.

We emphasize that if E_p or E'_p is empty, then Δ_p does *not* play a role in $\Lambda(n) \circ \Lambda(n')$, and we assume that $\Delta_p(n \to n') = [0, \infty)$ in such a case, as in [2, 1, 9]. It follows that for $n \to n' \to n''$, $(\Lambda(n) \circ \Lambda(n')) \circ \Lambda(n'')$ is the same as $\Lambda(n) \circ (\Lambda(n') \circ \Lambda(n''))$. Thus, we can unambiguously define the TC-MSC T^ρ *induced* by a path $\rho = n_0 \dots n_\ell$ of G to be $\Lambda(n_0) \circ \dots \circ \Lambda(n_\ell)$.

From now on, whenever there is no confusion, we shall speak interchangeably of a node n and its associated TC-MSC $\Lambda(n)$. We write $\mathcal{L}(G)$ for the union of $\mathcal{L}(T^\rho)$, where ρ ranges over final paths of G. We call a dated MSC in $\mathcal{L}(G)$ a *timed execution* of G.

Figure 1 provides an example of a TC-MSC graph G_1 and an TC-MSC T_1 induced by the path $n_1 \cdot n_1 \cdot n_2$ of G_1, i.e., $T_1 = T^{n_1 \cdot n_1 \cdot n_2}$. Further M_1 is a dated MSC generated by T_1 (i.e., it is a timed execution of G_1). As n_2 is final, $M_1 \in \mathcal{L}(G)$.

The emptiness problem for TC-MSC graphs is: given a TC-MSC graph G, determine whether $\mathcal{L}(G) = \emptyset$. We say that a path ρ is consistent whenever $L(T^\rho) \neq \emptyset$. Hence $\mathcal{L}(G) \neq \emptyset$ is equivalent with G having at least one consistent path. This is one of the fundamental verification problems that must be addressed on scenario based descriptions. Indeed, a TC-MSC graph with an empty language should be considered as ill-specified and it is helpful to catch this exception at an early stage of design. Unfortunately, in [9] it is shown that this problem (as well as checking reachability, boundedness) is undecidable in general.

In the rest of the paper, we show that checking emptiness for TC-MSC graphs is decidable under a restriction on time constraints, namely drift-boundedness, as defined below. Furthermore, we show that one can test whether a given TC-MSC graph satisfies this condition, thus providing an effective decidability procedure.

3 Drift-Boundedness

In this section we define the crucial notion of drift-boundedness. A TC-MSC graph G is said to be K-drift-bounded for some integer $K > 0$ if for all paths ρ of G, there exists *some* timed execution of T^ρ such that every TC-MSC labeling a node of ρ takes at most K units of time to complete. In other words, the dated MSC associated to the execution assigns dates such that for each instance of a node, the duration between the first and last date of (events in) this instance is at most K.

To define this more formally, let us start by fixing a TC-MSC graph G. Let $\rho = n_0 \dots n_\ell$ be a consistent path of G and $(E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda, d)$ be a dated MSC generated by T^ρ. For an integer K, we say that $(E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda, d)$ is a *K-drift-bounded* dated MSC of ρ iff for each $i = 0, \dots, \ell$, for any two events e, e' in $\Lambda(n_i)$, it is the case that $|d(e) - d(e')| \leq K$. We say that ρ is *K-drift-bounded* iff there *exists* a K-drift-bounded dated MSC in $\mathcal{L}(T^\rho)$. We emphasize that $\mathcal{L}(T^\rho)$ may also contain dated MSCs which are not K-drift-bounded. We say that G is K-drift-bounded iff every *consistent* (but not necessarily final) path of G is

K-drift-bounded. At last, G is drift-bounded iff it is K-drift-bounded for some K.

As an example, consider the TC-MSC graph G_1 from the figure above. G_1 is 2-drift-bounded since in every timed execution, we can be sure that all events in node n_1 or n_2 will be completed with a delay of at most 2 time units. But if we change the constraints on the loop on n_1 from $([0,1]_r, [0,1]_s)$ to, for instance, $([4,5]_r, [1,2]_s)$ then it is no longer k-drift-bounded for any integer k. Note that G_1 is not locally synchronized (as defined in [1] by lifting the same definition from the untimed setting [14, 4]). In fact, we can simulate the producer-consumer protocol on node n_1 and obtain non-regular behaviours. Thus, this example cannot be handled by [1] or other existing results in the timed setting.

We believe that drift-boundedness is a practical notion. Interpreting a node of a TC-MSC graph as a phase or a transaction of a distributed protocol, we expect any scenario labeling the node to be performed in a bounded time, say K. A protocol specified as a TC-MSC graph that is not K-drift-bounded should thus be considered as ill-formed. Indeed, while a TC-MSC graph specification is usually incomplete (as it abstracts away some events and constraints used in the actual implementation), if it forces time to drift, then every implementation of this specification will have non K-drift-bounded executions.

3.1 The main results

We can now state our main results. The first result establishes the decidability of the emptiness problem for K-drift-bounded TC-MSC graphs.

▶ **Theorem 5.** *Let G be a K-drift-bounded TC-MSC graph. Then checking whether $\mathcal{L}(G)$ is empty is decidable.*

An immediate question which arises is whether the drift-boundedness hypothesis of Theorem 5 can be effectively checked. Our second result shows that this is indeed decidable:

▶ **Theorem 6.** *Given a TC-MSC graph G and an integer K, one can effectively decide whether G is K-drift-bounded.*

Note that the decidability result in Theorem 6 is in fact at the boundary of undecidability. The definition of K-drift-bounded considers every path of a TC-MSC graph, including paths that cannot be extended to consistent final paths. Instead, if we consider the problem of checking whether every consistent *final* path of a TC-MSC graph is K-drift-bounded, this turns out to be undecidable. This fact can be shown by a simple reduction from the reachability problem shown to be undecidable in [9]. That is,

▶ **Proposition 7.** It is undecidable, given a TC-MSC graph G and an integer K, to determine whether every consistent final path of G is K-drift-bounded.

Proof. Given a TC-MSC graph G and an integer K, we reduce the reachability problem on G to the emptiness problem on another TC-MSC graph G' computed from G as follows. Recall that that in general one can not decide whether there exists some consistent path ρ that ends in node n, as shown in [9]. Consider a TC-MSC graph G with a single final node n_f. The undecidability result of [9] still holds in general for this kind of TC-MSC graph. Now, consider a TC-MSC T that has a constraint of the form $(K+1, \infty)$, that is, this TC-MSC is not K-drift-bounded. Now let us build a TC-MSC graph G' by adding a new node n_{new} to G, labeled by T, a transition without timing constraints from n_f to n_{new}, and setting n_{new} as only final node of G'. This way, every final path of G' ends at node n_{new}, and a consistent final path of G' (if one exists) can not be K-drift-bounded. Then, every consistent final path of G' is K-drift-bounded if and only if n_f is not reachable in G. ◀

We show first that one can assume the TC-MSC graph to be *full* as defined below, while preserving the two crucial properties that we consider in this paper, namely non-emptiness and drift-boundedness. This will greatly simplify our main proofs.

3.2 Full TC-MSC Graphs

Recall that a TC-MSC graph G is full iff for each process $p \in \mathcal{P}$, each node of G has at least one event on p. We prove now that given a TC-MSC graph G, we can "augment" G to obtain a full TC-MSC graph G'. We have $\mathcal{L}(G') \neq \emptyset$ iff $\mathcal{L}(G) \neq \emptyset$, and G' is drift-bounded iff G is.

To avoid clutter, we assume here that a (TC) MSC may contain events representing internal actions. For our purpose, it suffices to introduce an action symbol a and allow events to be labelled with $p(a)$ representing p's performing the action a. We set the location of $p(a)$ to be p. In the absence of internal actions, each $p(a)$ event can be simulated by introducing an auxiliary process $\hat{p}$, by creating a p-event e_p labelled $p!\hat{p}$ and a matching $\hat{p}$-event $e_{\hat{p}}$ labelled $\hat{p}?p$, and asserting that relative difference of dates of e_p, $e_{\hat{p}}$ should be in the singleton interval $[0, 0]$.

We obtain G' by applying the following modifications to $G = (N, \mathcal{T}, \Lambda, n_{in}, N_{fi}, \longrightarrow, \Delta)$. For each TC-MSC $T = (E, (<_p)_{p\in\mathcal{P}}, \mu, \lambda, \delta)$ in $\mathcal{T}$, we modify E by adding a new event e_p with $\lambda(e_p) = p(a)$ for each process p such that $E_p = \emptyset$. We thus obtain a new TC-MSC $T' = (E', (<_p)_{p\in\mathcal{P}}, \mu, \lambda', \delta)$. We leave every $<_p$ and δ unchanged, and thus e_p is an isolated point in the partial order $\leq$. Such an event e_p is said to be *dummy*, and the other events that already appeared in T are called *ordinary*. We keep $\Delta(n{\to}m)$ unchanged for each transition $n{\to}m$. Recall that for each transition (n, m) in G, if prior to adding the dummy events, either n or m has no p-event, then $\Delta_p(n, m) = [0, \infty)$. We call G' the *augmented* TC-MSC graph of G. Clearly, G' is full.

For a node n in G, refer to its corresponding node in G' as n'. For a path $\rho = n_0 \dots n_\ell$ in G, refer to its corresponding path in G' as $\rho' = n'_0 \dots n'_\ell$. With these notations , we have the following proposition:

▶ **Proposition 8.** $\mathcal{L}(G) = \emptyset$ iff $\mathcal{L}(G') = \emptyset$. Futhermore, if G is K-drift-bounded, then G' is $\widehat{K}$-drift-bounded, where $\widehat{K} = (|\mathcal{P}| - 1) \cdot K$.

Clearly, if $\mathcal{L}(G') \neq \emptyset$, then $\mathcal{L}(G) \neq \emptyset$ as any consistent path $\rho' = n'_0 \dots n'_\ell$ of G' corresponds to a path ρ in G that is also consistent, since one can obtain a dated MSC for ρ by deleting dummy events from a dated MSC by ρ'. If $\mathcal{L}(G) \neq \emptyset$, then taking a consistent path ρ of G and a dated MSC M, and the associated path ρ' of G', one can create a dated MSC $M' \in \mathcal{L}(G')$ labeling ρ' from M by adding the dummy events and setting the date of a dummy event e on p to be the same as the date of the event on p immediately before e (or date 0 if there is no such event).

The second part of proposition 8 follows from the following technical lemmas. A transition $n{\to}m$ of G is said to be *scenario-connected* iff for some process p, both n, m have events on process p.

▶ **Lemma 9.** *Suppose $\rho = n_0 \dots n_\ell$ is a path of G such that $n_h{\to}n_{h+1}$ is scenario-connected for every $h = 0, \dots, \ell-1$. Let $(E, (<_p), \mu, \lambda, d)$ be a dated MSC generated by ρ. Then for any indices i, j with $0 \leq i < j \leq \ell$, if e is an event in n_i, f an event in n_j, then $d(e) - d(f) \leq \widehat{K}$.*

Proof. It follows that one can choose a sequence of processes $p_i \dots p_{j-1}$, such that for each $h = i, \dots, j-1$, n_h, n_{h+1} both have events on process p_h. From the sequence $p_i \dots p_{j-1}$, we pick a subsequence $p_{\alpha_1} \dots p_{\alpha_z}$, where $z \leq |\mathcal{P}|$, as follows. Firstly, let α_1 be the largest index in $\{i, \dots, j-1\}$, such that $p_{\alpha_1} = p_i$. That is, $p_h \neq p_i$ whenever $\alpha_1 < h \leq j-1$.

Secondly, inductively, for $u = 1, \ldots$, suppose $\alpha_1, \ldots, \alpha_u$ have been set. Pick α_{u+1} to be the largest index in $\{\alpha_u + 1, \ldots, j-1\}$ such that $p_{\alpha_{u+1}} = p_{\alpha_u+1}$. That is, $p_h \neq p_{\alpha_u+1}$ whenever $\alpha_{u+1} < h \leq j-1$. It follows that $p_{\alpha_1}, p_{\alpha_2}, \ldots$, are pairwise distinct, and thus this procedure of picking indices $\alpha_1, \alpha_2, \ldots$ will terminate after picking $\alpha_z = j-1$ for some $z \leq |\mathcal{P}|$. We emphasize that $p_{\alpha_{u+1}} = p_{\alpha_u+1}$ for $u = 1, \ldots, z-1$.

Now for $h = 1, \ldots, z-1$, $p_i, \ldots, p_{j-1}$ pick events x_h, y_h from node n_{α_h+1} such that x_h is on process p_{α_h} and y_h is on process p_{α_h+1}. Further, pick event y_0 on process p_i from n_i and event x_z on process p_{j-1} from n_j. Existence of x_h, y_h, $h = 1, \ldots, z-1$, and y_0, x_z is guaranteed by construction of the sequence $p_i, \ldots, p_j$. Set $x_0 = e$ and $y_z = f$. For $h = 0, \ldots, z-1$, since y_h, x_{h+1} are of the same process, we have $d(y_h) \leq d(x_{h+1})$. Since $(E, (<_p), \mu, \lambda, d)$ is K-drift-bounded, we have $d(x_h) - d(y_h) \leq K$ for $h = 0, \ldots, z$. Suppose e is on process p_e and f on process p_f. Recall that $p_{\alpha_1}, \ldots, p_{\alpha_z}$ are pairwise distinct. We show $d(e) - d(f) \leq \widehat{K}$ by considering four cases.

—Case (1). If $p_e, p_{\alpha_1}, \ldots, p_{\alpha_z}, p_f$ are pairwise distinct, then $z \leq |\mathcal{P}| - 2$, and thus $d(e) - d(f) \leq \sum_{h=0}^{z}(d(x_h) - d(y_h) + \sum_{h=0}^{z-1}(d(y_h) - d(x_{h+1})) \leq (z+1) \cdot K \leq \widehat{K}$.

—Case (2). If $p_e = p_{\alpha_t}$ for some t in $\{1, \ldots, z\}$ and $p_{\alpha_1}, \ldots, p_{\alpha_z}, p_f$ are pairwise distinct, then $z \leq |\mathcal{P}|-1$ and thus $d(e)-d(f) \leq d(e)-d(x_t)+\sum_{h=t}^{z} d(x_h)-d(y_h) \leq (z-t+1)\cdot K \leq \widehat{K}$.

—There remains two cases: (i) p_e, $p_{\alpha_1}, \ldots, p_{\alpha_z}$ are distinct, $p_f = p_{\alpha_t}$ for some t in $\{1, \ldots, z\}$. (ii) $p_e = p_{\alpha_t}$, $p_f = p_{\alpha_u}$ for some $t, u \in \{1, \ldots, z\}$. Both cases can be handled similarly to cases (1) and (2). ◀

Suppose $\rho = n_0 \ldots n_\ell$ is a path of G, and $(E, (<_p), \mu, \lambda, d)$ a dated MSC generated by ρ. For an integer C, we say $(E, (<_p), \mu, \lambda, d)$ is *C-distant* iff for any i, j in $\{0, \ldots, \ell\}$ with $i < j$, for any event e in n_i, f in n_j, it is the case that $d(e) - d(f) \leq C$. Note that unlike K-drift-boundedness, the notion of being C-distant places restriction on dates of events in two different nodes. Intuitively, being C-distant means if event e is at node which occurs earlier than the node in which event f is in, then e can be executed at most C time units later than f.

▶ **Lemma 10.** *Suppose that ρ is a K-drift-bounded consistent path of G. Then there exists a $\widehat{K}$-distant K-drift-bounded dated MSC generated by ρ.*

Proof. Let $(E, (<_p), \mu, \lambda, d)$ be a K-drift-bounded dated MSC generated by ρ. Let $\rho = n_0 \ldots n_\ell$. If $n_h \to n_{h+1}$ is scenario-connected for every $h = 0, \ldots, \ell - 1$, then $(E, (<_p), \mu, \lambda, d)$ is $\widehat{K}$-distant. Now suppose such is not the case. Let $t_1, \ldots, t_z$ be all the indices in $\{0, \ldots, \ell-1\}$ such that $n_{t_i} \to n_{t_i+1}$ is not scenario-connected, where $t_1 < \ldots < t_z$. It follows from the proof of lemma 9 that if e is an event in n_i, f an event in n_j, and none of $t_1, \ldots, t_z$ falls within $\{i, \ldots, j-1\}$, then $d(e) - d(f) \leq \widehat{K}$.

Observe that for each $i = 1, \ldots, z$, there is no time constraint dictated between an event in $n_0, \ldots, n_{t_i}$ and an event in $n_{t_i+1}, \ldots, n_\ell$. Fix an integer c whose choice is to be determined later. From $(E, (<_p), \mu, \lambda, d)$, we construct a new dated MSC $(E, (<_p), \mu, \lambda, d')$ by inductively applying the modifications associated with $t_1, \ldots, t_z$ as follows. Firstly, we apply the modification associated with t_1, which is to add c to the date of each event in $n_{t_1+1}, \ldots, n_\ell$ (while the date of any event in $n_0, \ldots, n_{t_1}$ remains unchanged). Inductively, suppose modifications associated with $t_1, \ldots, t_{i-1}$ have been done, for some $i \leq z$. We further apply the modification associated with t_i, which is to add c to the date of each event in $n_{t_i+1}, \ldots, n_\ell$ (while the date of any event in $n_0, \ldots, n_{t_i}$ remains unchanged).

Note that the date of an event is non-negative, by choosing c such that $d(g) - \widehat{K} \leq c$ for every event g in $n_0 \ldots, n_\ell$, one conclude that in $(E, (<_p), \mu, \lambda, d')$, for any event e in n_i, f in n_j, with $i < j$, and some of the indices $t_1, \ldots, t_z$ fall within $\{i, \ldots, j-1\}$, we have

$d'(e) - d'(f) \leq d(e) - c \leq \widehat{K}$. If none of the indices $t_1, \ldots, t_z$ falls within $\{i, \ldots, j-1\}$, then $d'(e) - d'(f) = d(e) - d(f) \leq \widehat{K}$ as observed earlier, following the proof of lemma 9. Clearly, $(E, (<_p), \mu, \lambda, d')$ is K-drift-bounded and fulfills the time constraints in ρ, since $(E, (<_p), \mu, \lambda, d)$ is a K-drift-bounded dated MSC generated by ρ. This completes the proof. ◀

The above lemma shows that K-drift-bounded and $\widehat{K}$-distant dated MSCs can be transformed into $\widehat{K}$-drift-bounded dated MSCs obtained by composition of full TC MSCs. Together with lemma 10, one establishes that, if G is K-drift-bounded, then G' is $\widehat{K}$-drift-bounded.

▶ **Lemma 11.** *Assume that there exists a dated MSC generated by a consistent path ρ of G, which is $\widehat{K}$-distant and K-drift-bounded. Then one can construct a $\widehat{K}$-drift-bounded dated MSC generated by ρ', the path in G' which corresponds to ρ.*

Proof. Let $\rho = n_0 \ldots n_\ell$, and let $M = (E, (<_p), \mu, \lambda, d)$ be a $\widehat{K}$-distant K-drift-bounded dated MSC generated by ρ. Recall the construction of G' from the beginning of section 3.2. We shall extend M to be to a dated MSC $M' = (E', (<_p), \mu, \lambda, d')$ generated by $\rho' = n'_0 \ldots n'_\ell$ as follows. Firstly, E' consists of events in ρ'. Secondly, we keep dates of events in E unchanged (that is, $d'(e) = d(e)$ for every $e \in E$), and assign suitable dates to dummy events. The assignment of dates to dummy events are done inductively, node by node, for nodes n_0, …, n_ℓ. Through the rest of this proof, for each $i = 0, \ldots, \ell$, pick an event f_i^{max} in n_i which has maximum date among events in n_i.

For node n_0, for any dummy e in n'_0, we set $d'(e) = max\{d(f_0^{max}) - \widehat{K}, 0\}$. Inductively, assume that dummy events in $n'_0, \ldots, n'_{i-1}$ have been assigned dates, then for any dummy event e in n'_0, we set $d'(e)$ to be the larger of $d'(e_{i-1})$ and $d(f_i^{max}) - \widehat{K}$, where e_{i-1} is the maximal event in n'_{i-1} which is on the same process as e. Note that e_{i-1} exists as n'_{i-1} is full.

Since ordinary events in M' has the same dates as in M, to see that M' satisfies the time constraints in ρ', it suffices to show:

Claim (1): For any $i = 0, \ldots, \ell - 1$, for any process p, if at least one of n'_i, n'_{i+1} contains a dummy event on p, then $d'(e_i) \leq d'(e_{i+1})$ where e_i is the maximal event on p in n'_i, and e_{i+1} the minimal event on p in n'_{h+1}.

We now prove Claim (1). Fix i,p. If e_{i+1} is a dummy event, then by definition of $d'(e_{i+1})$, we have $d'(e_i) \leq d'(e_{i+1})$. It remains to consider the case that e_i is a dummy event but e_{i+1} is not a dummy event. Let j be the largest index such that $0 \leq j < i$ and n'_j contains ordinary events on p. If such a j exists, set $D = d(e_j)$ where e_j is the maximal event on p in n'_j (which is an ordinary event); if no such j exists, set $j = -1$ and $D = 0$. By "unrolling" the definition of $d'(e_i)$, one sees that $d'(e_i)$ is the maximum in the set consisting of D and $d(f_h^{max}) - \widehat{K}$ for all indices h with $j < h \leq i$. Since e_{i+1} is on p, the choice of D ensures that $D \leq d(e_{i+1})$. Owing to that ρ is $\widehat{K}$-distant, we have $d(f_h^{max}) - \widehat{K} \leq d(e_{i+1})$ whenever $j < h \leq i$. These yield that $d'(e_i) \leq d(e_{i+1}) = d'(e_{i+1})$. —**End of proof of Claim (1)**

Having shown that M' is a dated MSC generated by ρ', we next prove that M' is $\widehat{K}$-drift-bounded. Since M is K-drift-bounded and $K \leq \widehat{K}$, it suffices to show:

Claim (2): For nodes n'_0, …, n'_ℓ in ρ', if e, g are events in n'_i such that at least one of e, g is a dummy event, then $|d'(e) - d'(g)| \leq \widehat{K}$.

We prove Claim (2) by induction on i. For $i = 0$, let e, g be events in n'_0 such that at least one of them is a dummy event. Say e is dummy. If g is also dummy, then $d'(g) = d'(e)$, else $d'(e) = max\{d(f_0^{max}) - \widehat{K}, 0\}$, $d'(g) = d(g)$ and $d(f_0^{max}) - K \leq d(g) \leq d(f_0^{max})$ would imply that $|d'(e) - d'(g)| \leq \widehat{K}$.

Assume now that Claim (2) holds for node $n'_0, \ldots, n'_{i-1}$. Let e, g be events in n'_i such that at least one of them is dummy. Say e is dummy. Let e_{i-1} (resp. g_{i-1}) be the maximal event in n'_{i-1} on the same process as e (resp. g).

—Case (1): g is not a dummy event.

If $d'(e) = d(f_i^{max}) - \widehat{K}$, then the same argument as in the base case of node n'_0 yields that $|d'(e) - d'(g)| \leq \widehat{K}$. Otherwise, we have $d'(e) = d'(e_{i-1})$. We have $d'(e) - d'(g) \leq d'(e_{i-1}) - d'(g_{i-1}) \leq \widehat{K}$ by induction hypothesis. And $d'(e) - d'(g) \geq (d(f_i^{max}) - \widehat{K}) - d(f_i^{max}) = -\widehat{K}$. These yield that $|d'(e) - d'(g)| \leq \widehat{K}$.

—Case (2): g is a dummy event.

If $d'(e) = d'(e_{i-1})$ and $d'(g) = d'(g_{i-1})$, then by induction hypothesis, we have $|d'(e) - d'(g)| \leq \widehat{K}$. The case of $d'(e) = d(f_i^{max}) - \widehat{K}$ and $d'(g) = d(f_i^{max}) - \widehat{K}$ is trivial. So it remains to consider the case that exactly one of $d'(e) = d'(e_{i-1})$, $d'(g) = d'(g_{i-1})$ holds. Since both e, g are dummy events, w.l.o.g. assume $d'(e) = d'(e_{i-1})$ but $d'(g) \neq d'(g_{i-1})$. That is, $d'(g) = d(f_i^{max}) - \widehat{K} > d'(g_{i-1})$. Thus, $d'(e) - d'(g) < d'(e_{i-1}) - d'(g_{i-1}) \leq \widehat{K}$ by induction hypothesis, and $d'(e) - d'(g) \geq 0$ by definition of $d'(e)$. These yield that $|d'(e) - d'(g)| < \widehat{K}$.

—**End of proof of Claim (2)**

From Claim (1)(2), one concludes that M' is a $\widehat{K}$-drift-bounded dated MSC generated by ρ'. ◀

Finally, we show that the second part of Proposition 8 follows from Lemma 10 and Lemma 11. Suppose G is K-drift-bounded, and let $\rho' = n'_0 \ldots n'_\ell$ be a consistent path of G'. By the arguments in the proof of the first part of Proposition 8, the corresponding path $\rho = n_0 \ldots n_\ell$ in G is consistent and thus K-drift-bounded. By Lemma 10, there exists a $\widehat{K}$-distant K-drift-bounded dated MSC generated by ρ. Hence by Lemma 11, one can construct a $\widehat{K}$-drift-bounded dated MSC generated by ρ'. That is, ρ' is $\widehat{K}$-drift-bounded. We have thus established that G is K-drift-bounded implies G' is $\widehat{K}$-drift-bounded.

4 Checking Emptiness of a K-Drift-Bounded TC-MSC Graph

Throughout the rest of the paper, we fix a TC-MSC graph G. To avoid clutter, we prove Theorems 5 and 6 in the special case where G is *full*, and constraints in G are only of the form $[a, b]$ and $[a, \infty)$. Extending proofs to handle constraints of the forms (a, b), $(a, b]$, $[a, b)$, (a, ∞) is straightforward and all statements hold in general, but additional notations are needed to remember whether each inequality is strict or not. Extending the proofs to handle a TC-MSC graph which is not full is done by augmenting it to a full TC-MSC graph and enlarging the bounds from K to $\widehat{K}$ (cf previous section).

We first describe intuitively the key ingredients of the proof of Theorem 5.

- The first observation is that checking consistency of a path ρ is equivalent to checking existence of a solution of a system of inequalities $\Phi(\rho)$, where each inequality is of the form $a \leq x_e - x_f$, where x_e, x_f are variables (ranging over $\mathbb{R}$) depicting the dates of events e, f of T^ρ and a is a (possibly negative) integer. Notice that constraints of the form $x_e - x_f \leq b$ can be written as $-b \leq x_f - x_e$.
- Next, observe that for M a dated MSC generated by a path ρ and n a node, checking whether M can be extended with n by assigning appropriate dates to events in n can be done with only information on the relative difference of dates of the last event of M on each process. This motivates us to associate to each path ρ a system $PF(\rho)$ of inequalities, called the *symbolic profile* of ρ, where each inequality is of the form $a_{pq} \leq x_p - x_q$, with each x_r being the date of the last event on $r \in \mathcal{P}$. Hence, each solution $(d_p)_{p \in \mathcal{P}}$ of $PF(\rho)$

corresponds to the dates of final events of some dated MSC generated by T^ρ, and vice versa. In particular, $PF(\rho)$ has a solution iff ρ is consistent.

- We call a_{pq} *coefficients*. We remark that coefficients can be chosen as integers. We finally restrict coefficients to be within $[-K, K]$. This does not hinder checking for emptiness when G is both full and K-drift-bounded. We can then represent $PF(\rho)$ in a finite way and build the set $\{PF(\rho) \mid \rho \text{ path of } G\}$ with a finite automaton.

4.1 Systems of Inequalities and Fourier-Motzkin elimination.

We first fix basic terminologies of systems of difference inequalities. Let X be a finite nonempty set of real-valued variables. A *(difference) inequality* is an inequality of the form $a \leq x - y$, where x, y are two different variables in X.

▶ **Definition 12.** A *system of (difference) inequalities* ϕ over X is $\wedge_{(x,y)\in R}\, a_{xy} \leq x - y$ where $R \subseteq X \times X$ is an irreflexive relation. We say that ϕ has *integral coefficients* whenever a_{xy} is a *(possibly negative) integer* for all $(x, y) \in R$.

Note that in this definition and later, we assume that the system is *simplified*, that is, for each $x, y \in X$, there is at most one inequality of the form $a \leq x - y$. This involves no loss of generality as $a \leq x - y \wedge a' \leq x - y$ is equivalent with $max(a, a') \leq x - y$.

If $a \leq x - y$ appears in ϕ, we say that ϕ contains an *edge* (x, y), and the weight of this edge is a. We say that two systems ϕ, ψ of inequalities are *equivalent* when ϕ has a solution in the real domain iff ψ has a solution in the real domain.

A key ingredient of our proofs is to propagate constraints concerning variables in a subset Y on variables in $X \setminus Y$ (depicting the dates of the last event of ρ on their process). Then the constraints on Y can be safely removed while keeping an equivalent system. This is done using the *Fourrier-Motzkin* elimination technique (see [7, 12]) described hereafter. Let $\phi = \{a_{ij} \leq x_i - x_j\}$ be a system of inequalities over a set of variabes X, and let $x \in X$ be a variable to eliminate from ϕ. We want to obtain a new system of inequalities ϕ' over variables $X \setminus \{x\}$ that is equivalent with ϕ.

Fourrier-Motzkin elimination technique: First, partition ϕ into three distinct systems of inequalities $\phi = \phi_1 \wedge \phi_2 \wedge \phi_3$, where ϕ_1 is the system of inequalities that do not involve x, ϕ_2 is the system of inequalities $\wedge_{i\in I}\, a_i \leq x - x_i$ that involve x as first operand, and ϕ_3 is the system of inequalities $\wedge_{j\in J}\, a_j \leq x_j - x$ that involve x as second operand. We have $\exists x \in \mathbb{R}, \phi_2 \wedge \phi_3$ is equivalent with $\exists x \in \mathbb{R}, max_{i\in I}((a_i + x_i)) \leq x \leq min_{j\in J}((x_j - a_j))$. We can thus eliminate variable x to obtain an equivalent formula $max_{i\in I}((a_i + x_i)) \leq min_{j\in J}((x_j - a_j))$. The inequality $max_{i\in I}((a_i + x_i)) \leq min_{j\in J}((x_j - a_j))$ is equivalent with the system of $|I| \times |J|$ inequalities $\psi = \wedge_{i\in I, j\in J}(a_j + a_i) \leq x_j - x_i$. Notice that if both a_i, a_j are integers, then so is $a_j + a_i$. That is, $\exists x \in \mathbb{R}, \phi$ is equivalent with the formula $\phi_1 \wedge \psi$ which does not contain variable x. Thus, ϕ and $\phi' = \phi_1 \wedge \psi$ are equivalent. Furthermore, if ϕ has *integral coefficients*, then ϕ' has *integral coefficients* too.

Note that this elimination is not just a simple projection on $X \setminus \{x\}$. It propagates constraints attached to x on remaining variables. Notice also that the number of inequalities of ϕ' is at most $(|X| - 1)^2$, after *simplification* of ϕ'.

We now extend elimination to sets of variables. Let ϕ be a system of difference inequalities over $X \cup Y$. Let ψ_1 and ψ_2 be two systems of inequalities over Y obtained from ϕ by repeatedly applying Fourier-Motzkin elimination of each variable in X, but where the order in which variables of X are eliminated is different. Then it is possible that $\psi_1 \neq \psi_2$. However, we have $Sol(\psi_1) = Sol(\psi_2)$, denoting by $Sol(\psi)$ the set of solutions of a system of inequalities ψ. We thus fix an order in which we eliminate variables till the end of the paper. For

$F \subseteq X$, let $\phi_{|F}$ denote the (unique) system of inequalities over variables F obtained by performing Fourier-Motzkin elimination of variables in $X \setminus F$ following the order previously fixed. Regardless of the order, ϕ and $\phi_{|F}$ are equivalent. Furthermore, if ϕ has *integral coefficients*, then so does $\phi_{|F}$.

4.2 Symbolic Profiles

In the rest of the paper, we will study systems of inequalities over variables that represent occurrence dates of events. Let E be a nonempty set of events. We denote by x_E the real-valued variable standing for the date of event $e \in E$, and let $X_e = \{x_e \mid e \in E\}$. Let $\rho = n_0 \dots n_\ell$ be a path of a TC-MSC graph G inducing TC-MSC $T^\rho = (E, \leq, \lambda, \delta)$. We associate path ρ with a system of linear inequalities $\Phi(\rho)$ *with integral coefficients* as follows:

▶ **Definition 13.** The system $\Phi(\rho)$ of inequalities associated with ρ is a the system of inequalities over the set of variables X_E such that, for any $e, f \in E$ with $e \lessdot f$,
- if $\delta(e, f) = [L, U]$, then $\Phi(\rho)$ contains both $L \leq x_f - x_e$ and $-U \leq x_e - x_f$;
- if $\delta(e, f) = [L, \infty)$, then $\Phi(\rho)$ contains $L \leq x_f - x_e$.

Further, $\Phi(\rho)$ contains no other inequalities.

As dates of events of a dated MSC generated by T^ρ correspond to a solution of $\Phi(\rho)$, ρ is consistent iff $\Phi(\rho)$ has a solution.

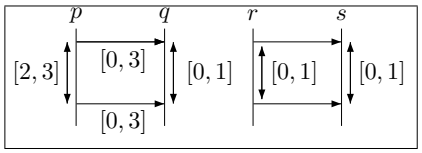
Let E be the set of events of T^ρ, and e_p be the last event of T^ρ on p, for all process p. Let E_{last} be the set $\{e_p \mid p \in \mathcal{P}\}$. Using Fourier-Motzkin elimination of variables $X' = \{x_e \mid e \notin E_{last}\}$, we obtain a system $\Phi|_{X_{last}}(\rho)$ over variables $X_{last} = \{x_e \mid e \in E_{last}\}$, with integral coefficients, equivalent with $\Phi(\rho)$. Once simplified, this system has at most $|\mathcal{P}|^2$ inequalities with integral coefficients. We encode this system as a *symbolic profile*.

▶ **Definition 14.** A *symbolic profile* σ is a function from $\mathcal{P} \times \mathcal{P}$ to $\mathbb{Z} \cup \{-\infty\}$. We denote by $\mathcal{PF}$ the (infinite) set of all profiles.

The symbolic profile $PF(\phi)$ induced by a system ϕ of inequalities over $X_{last} = \{x_p \mid p \in \mathcal{P}\}$ with integral coefficients, is defined by $PF(\phi)[p, q] = a_{pq}$ if $a_{pq} \leq x_p - x_q$ belongs to ϕ, and $PF(\phi)[p, q] = -\infty$ otherwise. We abusively use $PF(\phi)$ as a system of inequalities in the following, and denote x_p for x_{e_p}. Intuitively, $PF(\phi)_\rho[p, q] = -\infty$ means that there is no equation of the form $a_{pq} \leq x_p - x_q$ in ϕ. Recall that a system of inequalities is always simplified, that is it contains at most one inequality $a_{xy} \leq x - y$ for each pair of variables (x, y). For a path ρ, we denote $PF(\rho) = PF((\Phi(\rho))|_{X_{last}})$. We say that a symbolic profile $\sigma \in \mathcal{PF}$ is *satisfiable* if it has a solution, that is if $Sol(\sigma) \neq \emptyset$. It is easy to check if $PF(\rho)$ is satisfiable, for instance by using once again quantification elimination. As elimination preserves satisfiability, we have:

▶ **Proposition 15.** $PF(\rho)$ is satisfiable iff ρ is consistent.

Let us illustrate these notions through an example. Consider the TC-MSC graph G_1 from Figure 1 and consider the path $n_1 \cdot n_1$ which generates the TC-MSC $T^{n_1 \cdot n_1}$ as shown in Figure 2. Let e^i_j denote the i^{th} event on process j and E be the set of events of $T^{n_1 \cdot n_1}$. We obtain $\Phi(n_1 \cdot n_2)$ to be the set of inequalities over $X = \{x_e \mid e \in E\}$, where for instance the inequations $2 \leq x_{e^2_p} - x_{e^1_p}$ and $-3 \leq x_{e^1_p} - x_{e^2_p}$ capture the timing constraint $[2, 3]$ between e^1_p and e^2_p. Now we eliminate the variables $x_{e^1_p}, x_{e^1_q}, x_{e^1_r}, x_{e^1_s}$ to obtain a set of equations on $X_{last} = \{x_{e^2_p}, x_{e^2_q}, x_{e^2_r}, x_{e^2_s}\} = \{x_p, x_q, x_r, x_s\}$. Figure 2 depicts the resulting system of equations. For instance, $PF(n_1 \cdot n_1))[p, q] = \max(-3, -1 + 2 - 3) = -2$. The system of equations has many solutions.



$= T^{n_1 \cdot n_1}, \quad \phi(n_1 \cdot n_1)|_{X_{last}} = \left\{ \begin{array}{rcl} 0 & \leq & x_q - x_p \\ -2 & \leq & x_p - x_q \\ 0 & \leq & x_s - x_r \end{array} \right\}$

Figure 2 The TC-MSC induced by path $n_1 \cdot n_1$ of G_1 and its profile

We now explain how to compute $PF(\rho)$ in an inductive way, by defining an extension function $\theta^{n^- \to n} : \mathcal{PF} \to \mathcal{PF}$ for all transitions $n^- \to n$. For a symbolic profile σ and a transition $n^- \to n$, we define the profile $\theta^{n^- \to n}(\sigma)$ as follows:

- Firstly, form the system $\Psi = \psi_\sigma \wedge \psi_{n^- \to n} \wedge \psi_n$ of inequalities over $X = \{x_p \mid p \in \mathcal{P}\} \cup \{x_e \mid e \in E_n\}$ (x_p represents the date of process p in σ, E_n events of T^n), where:
  - ψ_σ consists of $\sigma(p,q) \leq x_p - x_q$ for every $p, q \in \mathcal{P}$, such that $\sigma(p,q) \neq -\infty$.
  - $\psi_{n^- \to n}$ contains, for each p with $\Delta_p(n^- \to n) = [L, U]$, two inequalities $-U \leq x_p - x_{f_p}$ and $L \leq x_{f_p} - x_p$, where f_p is the first event of n on p. For each p with $\Delta_p(n^- \to n) = [L, \infty)$, $\psi_{n^- \to n}$ contains the inequality $L \leq x_{f_p} - x_p$.
  - ψ_n is $\Phi(n)$, the system of inequalities associated with the singleton path n.
- Secondly, perform Fourier-Motzkin elimination on Ψ to remove all variables but $\{x_{\hat{e}_p}\}_{p \in \mathcal{P}}$ where $\hat{e}_p$ is the last event in n on p. Denote by Π the resulting system (after simplication) of inequalities over $\{x_{\hat{e}_p} \mid p \in \mathcal{P}\}$. Set $\theta^{n^- \to n}(\sigma) = PF(\Pi)$.

▶ **Lemma 16.** *For a path ρ and a transition $n^- \to n$ where n^- is the last node of ρ, we have $Sol(PF(\rho \cdot n)) = Sol(\theta^{n^- \to n}(PF(\rho)))$.*

Proof. First, consider the system of inequalities $\phi(\rho \cdot n)$ on variables X_E associated with path $\rho \cdot n$. Let E_n be the set of events of T^n, $E_{last} = \{e_p \mid p \in \mathcal{P}\}$ with e_p the last event of ρ on p, and E_ρ be the set of events of T^ρ. Let X_ρ, X_n, X_{last} be the variables associated respectively with sets of events E_ρ, E_n, E_{last}. We partition $\phi(\rho \cdot n) = \phi_1 \wedge \phi_2$ with $\phi_1 = \wedge_{x_e, x_f \in R_1} a_{e,f} \leq x_e - x_f$, for $R_1 = X_\rho \times X_\rho$ and $\phi_2 = \wedge_{x_e, x_f \in R_2} a_{e,f} \leq x_e - x_f$, where $R_2 = \big(X_n \times (X_n \cup X_{last})\big) \cup \big((X_n \cup X_{last}) \times X_n\big)$. Notice that $\phi_2 = \psi_n \wedge \psi_{n^- \to n}$.

Now, consider $\phi(\rho \cdot n)|_{X_n \cup X_{last}}$ where variables from $X_\rho \setminus X_{last}$ have been eliminated. This elimination keeps inequalities in ϕ_2 intact. That is, $\phi(\rho \cdot n)|_{X_n \cup X_{last}} = \phi_1|_{X_{last}} \wedge \phi_2$. Now, note that $\phi_1|_{X_{last}}$ precisely corresponds to $PF(\rho)$. That is, $\phi_1|_{X_{last}} = \psi_{PF(\rho)}$. Let us denote by X'_{last} the set of variables attached to the last events in $E_\rho \cup E_n$ Eliminating the variables from $X_n \cup X_{last} \setminus X'_{last}$, we get exactly $Sol(PF(\rho \cdot n)) = Sol(\theta^{n^- \to n}(PF(\rho)))$ (syntactically, the profiles may be different as the elimination orders may be different). ◀

Notice now that the set of profiles associated to paths of G is not finite in general. Else, by contradiction, a finite automaton could keep track of profiles of paths of G, and the emptiness problem would be decidable for any TC-MSC graph, a contradiction.

4.3 K-drift-bounded symbolic profiles

To obtain a finite set of symbolic profiles, we will use K-bounded symbolic profiles.

▶ **Definition 17.** A *K-drift-bounded symbolic profile* σ is a function from $\mathcal{P} \times \mathcal{P}$ to $\mathbb{Z} \cap [-K, K]$. We denote by $\mathcal{PF}_K$ the set of K-drift-bounded symbolic profiles.

The set $\mathcal{PF}_K$ is finite. To associate a K-drift-bounded symbolic profile to a path, we proceed as follows. First, we denote by $\Phi_K(\rho)$ the system of inequalities obtained from $\Phi(\rho)$ by the following modification: for each $i = 0, \ldots, \ell$, for any two different events e, f in the same

node n of ρ, if $\Phi(\rho)$ contains $a_{e,f} \leq x_e - x_f$, then replace it by $\max(a_{e,f}, -K) \leq x_e - x_f$; if $\Phi(\rho)$ does not have an edge (e, f), then add the inequality $-K \leq x_e - x_f$ (which is equivalent with $x_f - x_e \leq K$). Clearly, ρ is K-drift-bounded iff $\Phi_K(\rho)$ has a solution.

Now, similarly as the previous subsection, we can set $PF_K(\rho) = PF(\Phi_K(\rho)|_{X_{last}})$, for X_{last} the variable associated with the dates of last events of ρ on their process. By definition of $\Phi_K(\rho)$ and because G is full (this is crucial), we have $PF_K(\rho) \in \mathcal{PF}_K$.

▶ **Proposition 18.** ρ is K-drift-bounded and consistent iff $PF_K(\rho)$ is satisfiable.

The function $\theta^{n^- \to n}$ can be easily turned into an extension function $\theta_K^{n^- \to n}$ by adding the fact that the time difference between any pair of events in the node n is at most K.

▶ **Lemma 19.** *For a path ρ and a transition $n^- \to n$ where n^- is the last node of ρ, we have $Sol(PF_K(\rho \cdot n)) = Sol(\theta_K^{n^- \to n}(PF_K(\rho)))$.*

Notice that it is *not* the case that $PF_K(\rho)$ can be obtained from $PF(\rho)$ by setting $PF_K[p, q](\rho) = -K$ for all $PF[p, q](\rho) = a < -K$ and else $PF_K(\rho) = PF(\rho)$. This is because the K bound in $\phi_K(\rho)$ must be imposed on every node, not just the last one. Such constraints on past nodes can have implications for the profile of ρ.

4.4 Construction of a Symbolic Automaton

Let G be a K-drift-bounded full TC-MSC graph. We are now ready to construct a symbolic automaton $\mathcal{A}(G)$ such that $\mathcal{L}(G) \neq \emptyset$ iff $\mathcal{L}(\mathcal{A}(G)) \neq \emptyset$, thus completing the proof of Theorem 5.

- The states of $\mathcal{A}(G)$ are pairs (n, σ), with n a state of G and $\sigma \in \mathcal{PF}_K$.
- The initial state is $(n_{init}, PF_K(n_{init}))$,
- a state (n, σ) is final if n is final and σ is satisfiable.
- There is a transition from (n, σ) to (n', σ') iff there is a transition from n to n' and $\sigma' = \theta_K^{n \to n'}(\sigma)$.

We have easily that each path $n_1 \cdots n_z$ of G is associated with the path $(n_1, PF_K(n_1)) \cdots (n_z, PF_K(n_1 \cdots n_z))$ of $\mathcal{A}(G)$. We thus obtain easily:

▶ **Proposition 20.** Let G be a K-drift-bounded and full TC-MSC graph. Then $\mathcal{L}(\mathcal{A}(G)) = \emptyset$ iff $\mathcal{L}(G) = \emptyset$. Furthermore, $\mathcal{A}(G)$ has at most $|G| \times (2 \cdot K + 1)^{|\mathcal{P}|^2}$ states.

Compared with the related bibliography, we end up with an automaton much smaller in the worse case (exponential in $|\mathcal{P}|^2$ only, vs exponential in $|G|$ for [1]). Furthermore, being symbolic, the worse case is seldom reached, contrary to constructions of zones of timed automata accepting interleavings. Indeed, consider a path ρ made of one node, labeled by a TC-MSC with one event e_p for every $p \in \mathcal{P}$, and without constraints, hence allowing events to occur at any date. Without symbolic encoding, this path would give rise to $|2K|^{|\mathcal{P}|}$ states $(x_p)_{p \in \mathcal{P}}$, with $x_p \in \{0, (0, 1), 1, \cdots, K\}$ being the clock associated with e_p, for all $p \in \mathcal{P}$. On the other hand, we only keep the unique symbolic profile $PF(\rho)$ such that $\forall p, q \in \mathcal{P}$, $PF(\rho)[p, q] = -K$, meaning that $-K \leq x_p - x_q \leq K$, for all $p, q \in \mathcal{P}$.

5 Checking K-Drift-Boundedness of TC-MSC Graphs

The construction of automaton $\mathcal{A}(G)$ in Section 4 allows to decide for emptiness of $\mathcal{L}(G)$ under the hypothesis that G is K-drift-bounded. We show here that deciding whether G is K-drift-bounded is decidable, given K. The main idea is to look for a *minimal witness*: a

path $n_0 \dots n_z n_{z+1}$ of G is a *minimal witness* whenever $n_0 \dots n_{z+1}$ is consistent, $n_0 \dots n_z$ is K-drift-bounded but $n_0 \dots n_{z+1}$ is not.

▶ Remark. G is not K-drift-bounded iff there exists a minimal witness path in G.

We will search for a minimal witness using an automaton $\mathcal{B}(G)$ whose states have two components. The first component will test for K-drift-boundedness (which needs to hold for $n_0 \dots n_z$ but not for $n_0 \dots n_{z+1}$), and the second component will test for consistency (which should hold for $n_0 \dots n_{z+1}$). The first component keeps track of $PF_K(\rho)$, as in automaton $\mathcal{A}(G)$. It is sufficient as ρ is K-drift-bounded iff $PF_K(\rho)$ is satisfiable. Recall that without loss of generality, we assume G to be *full*. If it is not, the first component of $\mathcal{B}(G)$ should be adapted in a straightforward way. The second component keeps track of some $PF_{K_2}(\rho)$.

We first introduce a well known lemma to simplify the check for consistency, recalling that ρ is consistent iff the system of inequality associated with $PF(\rho)$ has a solution.

5.1 Shostak Lemma

Let φ be a (simplified) system of inequalities. A *cycle* in φ is a sequence $x_1 \dots x_m$ such that for all $i \in \{1, \dots, m-1\}$, $a_i \leq x_{i+1} - x_i$ appears in ϕ for some a_i, and $x_m = x_1$. The *weight* of this cycle is $\sum_{i \in \{1,\dots,m-1\}} a_i$. A cycle is simple when all variables are pairwise distinct, but the first and last one.

▶ **Proposition 21** (Shostak lemma [15])**.** φ has a solution iff every cycle in φ has weight at most zero iff every *simple* cycle in φ has weight at most zero.

We give here a proof in the special case where only $\leq$ is used in the system of inequalities ϕ. A complete proof for both $\leq$ and $<$ can be found in [15]

Proof to be adapted. First, assume that there exists $(X_i)_{i \in I}$ with $b_{i,j} \leq X_i - X_j$ for all $i, j \in \mathcal{P}$. By contradiction, assume that there exists a sequence $p_1 \cdots p_n$ of I^* with $p_n = p_1$ and $\sum_{1 \leq k < n} b_{p_k, p_{k+1}} > 0$. We have $0 = X_{p_1} - X_{p_n} = X_{p_1} - X_{p_2} + X_{p_2} \cdots - X_{p_n} \geq \sum_{1 \leq k < n} b_{p_k, p_{k+1}} > 0$, a contradiction.

We now prove by induction on the size of I that that if $\sum_{1 \leq k < n} b_{p_k, p_{k+1}} \leq 0$ for all sequence $p_1 \cdots p_n$ with $p_n = p_1$, then there exists $(X_i)_{i \in I}$ with $b_{i,j} \leq X_i - X_j$ for all $i, j \in I$. For $|I| = 1$, it is trivial. Let $I = \{1, \cdots, n\}$ be a set and $(b_{i,j})_{i,j \in I}$ be a set of constraints. If all constraints $b_{i,j}$ are negative, then it suffices to take $X_i = 0$ for all i. Else, we reschedule I such that $b_{1,2} > 0$. We let c be the max of $\sum_{1 \leq i < n} b_{p_i, p_{i+1}}$ over all sequences $p_1 \cdots p_n$ of I^* with $p_1 = 1$ and $p_n = 2$. In particular, $c \geq b_{1,2} \geq 0$ We fix $X_1 = X_2 + c$. We will then remove X_1 and all associated constraints $b_{1,i}, b_{i,1}$, and replace them by equivalent formula.

For that, we replace $b_{2,i}$ by $b'_{2,i} = max(b_{2,i}, b_{1,i} - c)$. Similarly, we replace $b_{i,2}$ by $b'_{i,2} = max(b_{2,i}, b_{1,i} + c)$, for all $i \geq 3$. We let $b'_{i,j} = b_{i,j}$ for all $i, j \geq 3$. We now prove that we still have $\sum_{1 \leq k < n} b'_{p_k, p_{k+1}} \leq 0$ for all sequence $p_1 \cdots p_n$ in $I \setminus \{1\}$ with $p_n = p_1$, in order to apply the induction hypothesis. Let $p_1 \cdots p_n$ be a sequence of $I \setminus \{1\}$ with $p_n = p_1$. There are two cases. First, if $p_i \neq 2$ for all i, then $\sum_{1 \leq k < n} b'_{p_k, p_{k+1}} = \sum_{1 \leq k < n} b_{p_k, p_{k+1}} \leq 0$, and we are done. Else, assume that $p_i = 2$ for a unique i (the case of several i is similar). We have $\sum_{1 \leq k < n} b'_{p_k, p_{k+1}} = \sum_{1 \leq k < i} b_{p_k, p_{k+1}} + \sum_{i+1 \leq k < n} b_{p_k, p_{k+1}} + b'_{p_{i-1}, 2} + b'_{2, p_{i+1}}$. Now, there are 4 cases for the value of $b'_{p_{i-1}, 2} + b'_{2, p_{i+1}}$.

1. $b'_{p_{i-1}, 2} = b_{p_{i-1}, 2}$ and $b'_{2, p_{i+1}} = b_{2, p_{i+1}}$: the sum is lower than 0 by hypothesis using the sequence $p_1 \cdots p_n$.
2. $b'_{p_{i-1}, 2} = b_{p_{i-1}, 1} - c$ and $b'_{2, p_{i+1}} = b_{1, p_{i+1}} + c$: the sum is lower than 0 as $c - c = 0$ and by hypothesis, using the sequence $p_1 \cdots \widehat{p_i} \cdots p_n$ where $\widehat{p_i} = 1$ instead of 2.

3. $b'_{p_{i-1},2} = b_{p_{i-1},1} - c$ and $b'_{2,p_{i+1}} = b_{2,p_{i+1}}$. Consider the sequence $2, p_{i+1}, \cdots p_n p_1 \cdots p_{i-1}, 1$. By adding and soustracting $b_{1,2}$ we get $\sum_{1\leq k<n} b'_{p_k,p_{k+1}} \leq -b_{1,2} - c \leq 0$ as both c and $b_{1,2}$ are positive.
4. $b'_{p_{i-1},2} = b_{p_{i-1},2}$ and $b'_{2,p_{i+1}} = b_{1,p_{i+1}} + c$: remember that $c = \sum_{1\leq i<n} b_{q_i,q_{i+1}}$ for some sequences $q_1 \cdots q_n$ of I^* with $q_1 = 1$ and $q_n = 2$. Consider the sequence $1, p_{i+1}, \cdots p_n p_1 \cdots p_{i-1} 2 q_2 \cdots q_n$. This is a sequence from 1 to $q_n = 1$, hence the sum of the associated b is lower or equal to 0 by hypothesis. Hence $\sum_{1\leq k<n} b'_{p_k,p_{k+1}} \leq 0$.

Now, by induction hypothesis, we can find $X_2, \cdots, X_n$ satisfying $(b'_{i,j})$. It is easy to see that fixing $X_1 = X_2 + c$ ensures $(b_{i,j})$. The only cases to look at is $X_1 - X_i = X_2 + c - X_i \geq b'_{2,i} + c \geq b_{1,i} - c + c = b_{1,i}$ and $X_i - X_1 = X_i - X_2 - c \geq b'_{i,2} - c \geq b_{i,1} + c - c = b_{i,1}$.

Notice that every cycle has weight at most zero iff every simple cycle has weight at most zero. Indeed, if we have a cycle with $x_i = x_j$, then one of the sum over $x_i \cdots x_j$ or of x_j to x_i going through $x_n = x_1$ would have a positive sum. Inductively using the argument to prune similar nodes, one ends up with a 'simple loop' with positive sum.

◀

5.2 Consistency Checking

Let $\rho = n_0 \dots n_{z-1}$ be a K-drift-bounded path of G and n_z a node in G such that $n_{z-1} \to n_z$. To check if $\rho \cdot n_z$ is consistent, we cannot just use $PF_K(\rho)$. This is because it is possible that $T^{\rho \cdot n_z}$ is consistent, but for every dated MSC M it generates, the part of M corresponding to T^ρ is not K-drift-bounded. Such possibilities must be accounted for. Instead, we use the fact that $\rho \cdot n_z$ is consistent is equivalent with $\phi(\rho \cdot n_z)$ has a solution, which is equivalent with every simple cycle in $\phi(\rho \cdot n_z)$ has weight at most zero, by Shostak Lemma.

We first define D to be the max over all transition $n \to n'$ of the sum of explicit (positive) lower bounds of time constraints appearing on this transition plus (positive) lower bounds of time constraints appearing in n'. Formally, we first fix a transition $n \to n'$. For all process $p \in \mathcal{P}$, let a_p be the lower bound of the interval $\Delta_p(n \to n')$. Also, for all events $e \lessdot f$ of n', we denote by $a_{e,f}$ the lower bound of the interval $\delta(e, f)$. Notice that $a_{e,f} \geq 0$ and $a_p \geq 0$ for all $p \in \mathcal{P}$ and $e \lessdot f$. We can now define $D_{n \to n'} = \Sigma_{e \lessdot f} a_{e,f} + \Sigma_{p \in \mathcal{P}} a_p$. Then we let $D = \max_{n \to n'} D_{n \to n'}$. At last, $K_2 = (|\mathcal{P}| + 1) \cdot K + D$. We now prove that:

▶ **Proposition 22.** Let $\rho \cdot n_z$ be a path of G such that $\rho = n_1 \cdots n_{z-1}$ is consistent and K-drift-bounded. Then $\Phi(\rho \cdot n_z)$ has a solution iff $\theta^{n_{z-1} \to n_z}(PF_{K_2}(\rho))$ is satisfiable.

As an automaton can maintain the information $PF_{K_2}(\rho)$ with a finite number of states, it can also test on the fly whether $\theta^{n_{z-1} \to n_z}(PF_{K_2}(\rho)) \neq \emptyset$, that is test whether $n_1 \cdots n_z$ is consistent, granted that $n_1 \cdots n_{z-1}$ is K-drift-bounded, which we can test.

Proof. We will consider three systems of inequalities.

The first one is $\phi_1 = \phi(\rho \cdot n_z)$. The second one, ϕ_2, is obtained from ϕ_1 by adding inequalities $-K_2 \leq x_e - x_f$ for all e, f from the same node of ρ.

We already know that ϕ_1 has a solution iff $PF(\rho \cdot n_z)$ is satisfiable iff $\theta^{n_{z-1} \to n_z}(PF(\rho))$ is satisfiable. It is easy to see that ϕ_2 has a solution iff $\theta^{n_{z-1} \to n_z}(PF_{K_2}(\rho))$ is satisfiable.

Hence, we just need to prove that ϕ_2 has a solution iff ϕ_1 has a solution to yield the statement of the proposition. Clearly, if ϕ_2 has a solution, then this solution is also a solution for ϕ_1. Conversely, assume that ϕ_1 has a solution. By Shostak lemma, it implies that every cycle in ϕ_1 has weight at most 0.

We want to prove that ϕ_2 has a solution. Again by Shostak lemma, it is sufficient to prove that every simple cycle of ϕ_2 has weight at most 0.

Let $x_1 \dots x_m$ be a simple cycle in ϕ_2. That is, for all $i \in \{1, \dots, m-1\}$, $b_i \leq x_{i+1} - x_i$ appears in ϕ_2 for some b_i, and $x_m = x_1$. We want to prove that $\sum_i b_i \leq 0$.

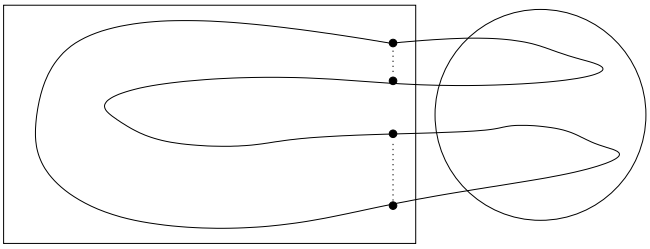
We now define the third system of inequalities: $\phi_3 = \phi_K(\rho)$. Notice that ϕ_3 can be obtained from ϕ_2 by deleting the events from n_z, and adding inequalities $-K \leq x_e - x_f$ for all e, f from the same node of ρ, as $K \leq K_2$. As ρ is K-drift-bounded and consistent, we know that ϕ_3 has a solution, that is every cycle in ϕ_3 has weight at most 0 by Shostak lemma.

Let a_i be the associated coefficients in ϕ_1 with $a_i \leq x_{i+1} - x_i$ (if a_i does not exists, fix $a_i = -\infty$) and c_i the associated coefficients in ϕ_3.

We have $a_i \leq b_i \leq c_i$ by definition of ϕ_1, ϕ_2, ϕ_3. First, if $a_i = b_i$ for all i, then the cycle $x_1 \dots x_m$ in ϕ_2 is also a cycle in ϕ_1 and $\sum_i b_i = \sum_i a_i$. As every cycle in ϕ_1 has weight at most 0, we are done.

Else, we have $a_k \neq b_k$ for some k. It means that $a_k < b_k$ and $b_k = -K_2$. Furthermore, e_k, e_{k+1} are in the same node n because ϕ_2 only adds constraints on pairs of events of the *same node* of ρ. Furhtermore $n \neq n_z$, which means that $c_k = -K > b_k$.

Now, consider I the set of indices i such that x_i or x_{i+1} belongs to n_z. Also, denote J the complementary set of indices j, that is such that x_j and x_{j+1} belongs to ρ. We have $\sum_i b_i = \sum_{i \in I} b_i + \sum_{i \in J} b_i$. Note that $k \in J$. We have $\sum_{i \in I} b_i \leq D$ by definition of D and because the cycle is simple. The set J is partitionned into pieces. Each piece $J' \subseteq J$ is made of "consecutive" indices, that is either $J' = \{i, i+1 \dots, j\}$ or $J' = \{i, \dots, m, 1, \dots, j\}$, such that $e_{i-1} \in n_z$ and $e_{j+1} \in n_z$. There are at most $|\mathcal{P}|$ pieces (because the cycle is simple). In the picture, there are 2 pieces. Furthermore, each piece begins and ends with an event of n_{z-1}. For all $j \in J$, we have $b_j \leq c_j$. We thus have $\sum_{j \in J} b_j = b_k + \sum_{j \in J \setminus \{k\}} b_j \leq b_k + \sum_{j \in J \setminus \{k\}} c_j \leq b_k - c_k + \left(\sum_{j \in J} c_j\right)$. Recalling that $\sum_{i \in I} b_i \leq D$, we have $\sum_i b_i \leq D + b_k - c_k + \left(\sum_{j \in J} c_j\right) \leq \left(\sum_{j \in J} c_j\right) - |\mathcal{P}| \cdot K$ as $b_k - c_k = -(|\mathcal{P}| \cdot K + D)$.



It now suffices to bound $\left(\sum_{j \in J} c_j\right)$. Recall that every cycle in ϕ_3 has weight at most 0. Let $J_1, \cdots, J_r$ be the pieces of J. recall that $r \leq |\mathcal{P}|$. For all $i \leq r$, denoting $J_i = \{s, \dots, t\}$, we rename $x_s \cdots x_t$ into $y^i_1 \cdots y^i_{n^i}$. We create the cycle $\xi = y^1_1 \cdots y^1_{n^1} \cdots y^r_1 \cdots y^r_{n^r} y^1_1$ by gluing all the variables together. Compared with $\sum_{j \in J} c_j$, for every $s \leq r$, we added an edge between $y^s_{n^s}$ and y^{s+1}_1. We have that both $e^s_{n^s}$ and e^{s+1}_1 are in n_{z-1}. In ϕ_3, there is an edge between any two events of the same node (and in particular in n_{z-1}), hence this connecting edge $c_s \leq y^{s+1}_1 - y^s_{n^s}$ exists, and $c_s \geq -K$, by definition of ϕ_3. Now, the sum over the cycle ξ in ϕ_3 is at most 0, that is $\left(\sum_{j \in J} c_j\right) \leq 0 - |\mathcal{P}|(-K) = |\mathcal{P}| \cdot K$. Hence $\sum_i b_i \leq 0$. ◀

5.3 The automaton construction

Now, writing the automaton $\mathcal{B}(G)$ to check whether G is K-drift-bounded is straightforward.

- The states of $\mathcal{B}(G)$ are triples (n, σ, τ), with n a state of G and $\sigma \in P_K$ and $\tau \in P_{K_2}$.
- The initial state is $(n_{init}, PF_K(n_{init}), PF_{K_2}(n_{init}))$,

- a state (n, σ, τ) is final if there exists a transition $n \to n'$ with:
  1. σ is satisfiable (ρ is K-drift-bounded)
  2. $\theta_K^{n \to n'}(\sigma)$ is not satisfiable ($\rho n'$ is not K-drift-bounded).
  3. $\theta^{n \to n'}(\tau)$ is satisfiable ($\rho n'$ is consistent).
- There is a transition from (n, σ, τ) to (n', σ', τ') iff σ' has a solution and there is a transition from n to n' and $\sigma' = \theta_K^{n \to n'}(\sigma)$ and $\tau' = \theta_{K_2}^{n \to n'}(\tau)$.

We thus obtain from Proposition 22, recalling $K_2 = (|\mathcal{P}| + 1) \cdot K + D$:

▶ **Proposition 23.** Let G be a full TC MSC graph. Then $\mathcal{L}(\mathcal{B}(G)) = \emptyset$ iff G is K-drift-bounded. Furthermore, $\mathcal{B}(G)$ has at most $|G| \times (2K+1)^{|\mathcal{P}|^2} \times (2K_2+1)^{|\mathcal{P}|^2}$ states.

Lifting the assumption that G is full is easy by working on the equivalent full TC-MSC graph with bound $\widehat{K}$ instead of K, and slightly changing the first component to check also that both the path in the full TCMSC graph is $\widehat{K}$-drift-bounded, and that the path in the original graph is K-drift-bounded.

6 Conclusion

This paper has addressed the emptiness problem for TC-MSC graphs. This problem was known to be decidable for locally synchronized TC-MSC graphs. This decision procedure relied on an expensive construction of a timed automaton recognizing timed linearizations followed by the construction of a standard zone automaton. We have shown that emptiness can be checked under the restriction that a TC-MSC graph is K-drift-bounded, for some K, together with the decidability of this restriction. The decision procedure does not consider linearizations of TC-MSC graphs. Instead, a finite automaton keeps track of a system of equations describing symbolically constraints over dates of last events on each process.

We will consider in the future how to use this symbolic representation when perfoming more involved verification on the TC-MSC graph (conformance of an implementation, logics).

References

1 S. Akshay, P. Gastin, M. Mukund, and K. Narayan Kumar. Model checking time-constrained scenario-based specifications. In *FSTTCS*, pages 204–215, 2010.

2 S. Akshay, M. Mukund, and K. Narayan Kumar. Checking coverage for infinite collections of timed scenarios. In *CONCUR'07*, pages 181–196, 2007.

3 R. Alur and D. L. Dill. A theory of timed automata. *TCS*, 126(2):183–235, 1994.

4 R. Alur and M. Yannakakis. Model checking of message sequence charts. In *CONCUR'99*, volume 1664 of *LNCS*, pages 114–129, 1999.

5 P. Bouyer, S. Haddad, and P.-A. Reynier. Timed unfoldings for networks of timed automata. In *ATVA'06*, volume 4218 of *LNCS*, 2006.

6 F. Cassez, Th. Chatain, and C. Jard. Symbolic unfoldings for networks of timed automata. In *ATVA'06*, volume 4218 of *LNCS*, pages 307–321, 2006.

7 G. Dantzig and B. Curtis Eaves. Fourier-motzkin elimination and its dual. *J. Comb. Theory, Ser. A*, 14(3):288–297, 1973.

8 C. Dima and R. Lanotte. Distributed time-asynchronous automata. In *ICTAC'07*, pages 185–200, 2007.

9 P. Gastin, M. Mukund, and K. Narayan Kumar. Reachability and boundedness in time-constrained MSC graphs. In *Perspectives in Concurrency Theory*, IARCS-Universities, pages 157–183. 2009.

10 J. G. Henriksen, M. Mukund, K. Narayan Kumar, M. Sohoni, and P. S. Thiagarajan. A theory of regular MSC languages. *Inf. and Comp.*, 202(1):1–38, 2005.

11 ITU-TS Recommendation Z.120: Message Sequence Chart 1999 (MSC99), 1999.

12 B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, Germany, 3rd edition, 2006.

13 D. Lugiez, P. Niebert, and S. Zennou. A partial order semantics approach to the clock explosion problem of timed automata. *TCS*, 345(1):27–59, 2005.

14 A. Muscholl and D. Peled. Message sequence graphs and decision problems on mazurkiewicz traces. In *MFCS'99*, volume 1672 of *LNCS*, pages 81–91, 1999.

15 R. Shostak. Deciding linear inequalities by computing loop residues. *JACM*, 28(4):769–779, 1981.