Variable-Delay Controllability

Nikhil Bhargava, Christian Muise, Brian Williams

Massachusetts Institute of Technology

{nkb, cjmuise, williams}@mit.edu

Abstract

In temporal planning, agents must schedule a set of events satisfying a set of predetermined constraints. These scheduling problems become more difficult when the duration of certain actions are outside the agent's control. *Delay controllability* is the generalized notion of whether a schedule can be constructed in the face of uncertainty if the agent eventually learns when events occur. Our work introduces the substantially more complex setting of determining *variable-delay controllability*, where an agent learns about events after some unknown but bounded amount of time has passed. We provide an efficient $O(n^3)$ variable-delay controllability checker and show how to create an execution strategy for variable-delay controllability problems. To our knowledge, these essential capabilities are absent from existing controllability checking algorithms. We conclude by providing empirical evaluations of the quality of variable-delay controllability results as compared to approximations that use fixed delays to model the same problems.

1 Introduction

In temporal planning problems, agents are required to find schedules for events that satisfy a set of predetermined constraints. This scheduling problem becomes more difficult when the duration of certain actions are outside of an agent's control (e.g. when driving to work, it is impossible to give a precise schedule because of the unpredictability of traffic).

When temporal planning problems include uncertainty, *delay controllability* checking provides a generalized way of determining the feasibility of the temporal plan, unifying the previous notions of strong and dynamic controllability [Bhargava *et al.*, 2018]. Informally, under delay controllability, the main actor learns about the true duration of uncertain events after some amount of time has elapsed, and delay controllability assesses the feasibility of the temporal plan under different observation patterns.

While delay controllability gives us a rich language with which to model temporal planning in the face of uncertainty, it still has some fundamental limitations. In particular, it requires that an agent either learns the exact time that an event

happens or learns nothing about the event. In contrast, there are many instances where some of the temporal ambiguity around an event is resolved after time passes even if it is impossible to learn its exact occurrence. For example, a human actor is unlikely to remember that they finished lunch at precisely 1:08pm several hours later, but they may be able to say that they finished lunch between 1:00pm and 1:15pm. Often, this partial reduction in uncertainty is both necessary and sufficient to ensure that the remaining constraints can be satisfied, but this uncertainty reduction might only happen well after the initial action occurred. Others have created models where the observation of events allows us to branch and consider distinct possible states [Moffitt and Pollack, 2007; Combi *et al.*, 2013], but these models do not consider how to incorporate partial resolution of uncertainty in events that have already happened. In this paper, we focus on addressing this modeling gap and provide three main contributions.

First, we provide a formalism for modeling uncertain observations of events. We build our model and controllability checking algorithms on top of Simple Temporal Networks with Uncertainty [Vidal and Fargier, 1999] and for each action with uncertain duration, we provide a bounded interval representing the amount of time that may pass after an event occurs before an agent learns that it occurs.

Second, we provide an efficient algorithm for verifying that these networks are controllable under uncertain observation. We approach the problem by creating a parallel temporal network that is controllable with respect to some fixed observations if and only if our original network is controllable. If we let n be the number of events in our schedule and m be the number of constraints, we can apply this transformation in $O(m+n)$ time, and since controllability checking under fixed observation can be performed in $O(n^3)$ time [Bhargava *et al.*, 2018], we can similarly check variable-delay controllability in $O(n^3)$ time. Along with a controllability check, we also show how to derive an execution strategy for controllable networks with uncertain observations. The same approach used to assess the controllability of the network yields a new, less expressive network that is controllable with respect to some fixed-delay observations. We show how only a few modifications are needed to reduce our original execution problem to that of finding an execution strategy on the less expressive network.

Third and finally, we provide an empirical characterization

of the quality of variable-delay controllability as contrasted against controllability checks that approximate the model using fixed delays. We show that the false positive rate is low but not zero, indicating that it is most appropriate to check variable-delay controllability directly.

Beyond temporal and observational uncertainty, many others have extended temporal networks and temporal network controllability checking to suit particular needs. Probabilistic Simple Temporal Networks ascribe probability distributions to the uncontrollable constraints of a network [Fang *et al.*, 2014], and Conditional Simple Temporal Networks with Uncertainty allow for conditional execution and enforcement of constraints based on the observation of pre-specified events [Combi *et al.*, 2013]. When new temporal constraints are added one at a time to a network, iterative controllability checks show improvements over non-iterative solutions [Stedl and Williams, 2005; Shah *et al.*, 2007; Nilsson *et al.*, 2013; 2014], and when constraints are instead iteratively relaxed, RelaxIDC [Bhargava *et al.*, 2017] gives us an algorithm that speeds up controllability checking over time.

Despite the many existing extensions to temporal networks and controllability checking, we argue that the ability to model, validate, and execute temporal networks under observational uncertainty represents a unique and significant improvement over state of the art. With previous temporal controllability formalisms, inference only flowed forward with time. Observations that happened in the future had no impact on our beliefs about past events. In contrast, our approach provides a rigorous means of incorporating future observations in our updated beliefs about past events. While others have worked on extending existing models to account for these relationships [Moffitt and Pollack, 2007; Moffitt, 2007; Bit-Monnot *et al.*, 2016; Casanova *et al.*, 2016], none have provided a sound and complete algorithm for verifying controllability in arbitrary networks.

2 Background & Definitions

In temporal planning, Simple Temporal Networks (STN) are composed of a set of events and simple binary constraints relating those events (e.g. event A must happen at least 20 minutes before event B) [Dechter *et al.*, 1991]. STNs are useful for modeling problems where the agent can precisely schedule the timings of events, but they do not let us model actions whose durations are uncertain.

Simple Temporal Networks with Uncertainty (STNU) augment the STN allowing us to model these types of uncertain actions [Vidal and Fargier, 1999]. Events are subdivided into *activated* and *received* timepoints while constraints are subdivided into *free* and *contingent* constraints.

Definition 1. STNU [Vidal and Fargier, 1999]
An STNU is a 4-tuple $\langle X_b, X_e, R_c, R_g \rangle$ where:

- X_b is the set of activated timepoints

- X_e is the set of received timepoints

- R_c is the set of free constraints of the form $l_c \leq x_i - x_j \leq u_c$, where $x_i, x_j \in X_b \cup X_e$

- R_g is the set of contingent constraints of the form $l_g \leq e_i - b_j \leq u_g$, where $e_i \in X_e$, $b_j \in X_b$

Activated timepoints represent events whose values are explicitly scheduled by the main agent while received timepoints are assigned values by some external actor. Free constraints behave like ordinary STNU constraints while contingent constraints represent actions with values of uncertain duration. Contingent constraints always begin with an activated timepoint, representing an initiated action, and end with a received timepoint.

Since an STNU has events whose occurrences are outside of an agent's control, to assess the feasibility of an STNU, we examine its delay controllability to determine whether a valid schedule can be constructed given a guarantee that future uncertainty will eventually be fully resolved [Bhargava *et al.*, 2018]. In the rest of this paper, we will refer to the original notion of delay controllability as *fixed-delay controllability* for clarity.

Fixed-delay controllability uses a *fixed-delay function* to parameterize controllability. The fixed-delay function encodes the amount of time that must pass before we observe a received timepoint when constructing our schedule.

Definition 2. Fixed-Delay Function [Bhargava *et al.*, 2018]
A fixed delay function, $\gamma : X_e \to \mathbb{R}^+ \cup \{\infty\}$ maps from received timepoints to the amount of time that must pass before that timepoint is observed.

Definition 3. Fixed-Delay Controllability [Bhargava *et al.*, 2018]
An STNU is fixed-delay controllable with respect to some delay function γ if it is possible for an agent to reactively execute a plan if they learn the true value of received timepoint e only after an additional $\gamma(e)$ time has passed.

Importantly, fixed-delay controllability generalizes other notions of controllability. For example, by using a fixed-delay function where we observe all timepoints, checking fixed-delay controllability reduces to checking *dynamic controllability*. Similarly, a fixed-delay function specifying that we never observe any received timepoints corresponds to checking *strong controllability* [Vidal and Fargier, 1999].

As a matter of convention, we will use $A \to B$ to represent requirement links between timepoints A and B and will use $A \Rightarrow E$ to represent contingent links between A and E. When we talk about the delay function associated with the received timepoint E of some contingent link $A \Rightarrow E$, we will use the notation $\gamma(e)$.

2.1 Variable-Delay Controllability

While fixed-delay controllability is quite expressive, its fundamental limitation is that events must be observed only after a fixed offset. If there were some uncertainty in our observation event, the controllability problem becomes much harder, as we now only have rough bounds around when an event in the past happened rather than ground truth. We now rigorously introduce the notions of a *variable-delay function* and *variable-delay controllability*.

Definition 4. Variable-Delay Function
A variable-delay function, $\bar{\gamma} : X_e \to (\mathbb{R}^+ \cup \{\infty\}) \times (\mathbb{R}^+ \cup$

$\{\infty\}$), takes a received timepoint and outputs an interval $[a, b]$. The range bounds the least amount of time and most amount of time that may pass after the assignment of a value to the received timepoint before a value is known to be assigned. By convention, we use $\bar{\gamma}^-(e)$ to represent the lower-bound in observation and $\bar{\gamma}^+(e)$ to represent the upper-bound.

Definition 5. Fixed Delay Set

We say $F(\bar{\gamma})$ represents the set of all possible fixed-delay functions that are valid groundings of the variable-delay function $\bar{\gamma}$. In other words, $\forall \gamma \in F(\bar{\gamma}), \forall e \in X_e : \gamma(e) \in [\bar{\gamma}^-(e), \bar{\gamma}^+(e)]$.

Definition 6. Possible Observed Situations

Let Ω be the set of all possible sets of assignments of durations to contingent links, let $\delta : X_b \to \mathbb{R}$ be a partially committed schedule for activated timepoints, let $\omega \in \Omega$ be the actual assignment of durations to contingent links, let $\bar{\gamma}$ be the variable-delay function, and let $\gamma \in F(\bar{\gamma})$ be the actual observational delay. The set of all possibly observed situations before t with respect to some $\delta, \omega, \bar{\gamma}, \gamma$ is $\Omega_{<t,\delta,\omega,\bar{\gamma},\gamma} = \{\omega' \in \Omega : p(\omega, \omega', \delta, t, \bar{\gamma}, \gamma) \land q(\omega, \omega', \delta, t, \bar{\gamma}, \gamma)\}$, where:

- $p(\omega, \omega', \delta, t, \bar{\gamma}, \gamma) = \bigwedge_{\omega_{ij} \in \omega} (\delta(b_i) + \omega_{ij} + \gamma(e_j) > t) \to (\delta(b_i) + \omega'_{ij} + \bar{\gamma}^+(e_j) > t)$; or for any event that has not yet been seen, it is possible to have not yet seen that event in ω'.

- $q(\omega, \omega', \delta, t, \bar{\gamma}, \gamma) = \bigwedge_{\omega_{ij} \in \omega} (\delta(b_i) + \omega_{ij} + \gamma(e_j) \le t) \to (\gamma(e_j) + \omega_{ij} - \omega'_{ij} \in \bar{\gamma}(e_j))$; or for any event that has been seen, it must be possible to see that event at the same moment in ω'.

Definition 7. Variable-Delay Controllability

For any δ, let $\delta_{<t,\delta}$ represent a restriction on the set of assignments δ, where $\delta_{<t,\delta}$ only contains assignments for events that happen before time t. An STNU S is variable-delay controllable with respect to some variable-delay function $\bar{\gamma}$ if and only if: $\forall \omega \in \Omega, \forall \gamma \in F(\bar{\gamma})$, there exists an assignment δ, such that $\forall(x, t) \in \delta, \forall \omega' \in \Omega_{<t,\delta,\omega,\bar{\gamma},\gamma}$, there exists δ', such that $\delta_{<t,\delta} \subseteq \delta'$ and δ' is an assignment that satisfies $S_{\omega'}$.

In other words, an STNU is variable-delay controllable with respect to some variable-delay function $\bar{\gamma}$ if it is always possible to construct a satisfying plan for the future given the inferences that have been made about events that have already been observed.

It is clear that variable-delay controllability is a generalization of fixed-delay controllability. For any fixed-delay function γ, we can produce a corresponding variable-delay function $\bar{\gamma}$ where $\bar{\gamma}^+(e) = \bar{\gamma}^-(e) = \gamma(e)$.

3 Determining Controllability

Our strategy for determining whether a given STNU S is variable-delay controllable with respect to $\bar{\gamma}$ is to instead construct a related STNU S' that is fixed-delay controllable with respect to some derived delay function γ'.

It is important to note that we do not expect S' and γ' to be expressive enough to model our original problem. Because our focus is on checking the controllability of our problem, it

suffices for us to focus on the worst-case areas of our problem, or the situations that are hardest to schedule, and we construct S' and γ' with that end in mind.

To construct S' and γ', we start by copying the original graph S. We then make a series of iterative modifications to our new outputs so that they capture the controllability of our original input. We start by updating our definition of γ' based on observed values of $\bar{\gamma}$.

Lemma 1. *For any received timepoint $e \in X_e$ in S, if $\bar{\gamma}^-(e) = \bar{\gamma}^+(e)$, we can express the same behavior in S' using $\gamma'(e) = \bar{\gamma}^+(e)$.*

Proof. If $\bar{\gamma}^-(e) = \bar{\gamma}^+(e)$, then $\bar{\gamma}$ already emulates a fixed-delay for timepoint e. Assigning $\gamma'(e) = \bar{\gamma}^+(e)$ makes no change to the proposed controllability since the modified S' and γ' cover the same scenarios as S and $\bar{\gamma}$. $\square$

Lemma 2. *If for some $e \in X_e$, $\bar{\gamma}^+(e) = \infty$, we can express the same behavior in S' using $\gamma'(e) = \infty$*

Proof. If $\bar{\gamma}^+(e) = \infty$, then in some scenarios, e is unobservable in S. It does not matter that in some instances we may learn about e after some delay; controllability checking is about verifying that a valid execution strategy exists in all scenarios. Thus, if we can verify that S' is controllable when e is unobservable (when $\gamma'(e) = \infty$), then we know that S is controllable whenever e is observed after $t \in \bar{\gamma}(e)$ since a valid execution strategy could always choose to ignore the observation. If S' is uncontrollable when $\gamma'(e) = \infty$, then we also similarly know that we would not be able to find a valid execution strategy if e ended up unobservable in S. $\square$

After repeated applications of Lemmas 1 and 2, we are left with a series of received timepoints whose variable-delay function values represent finite ranges. Since the delay in observation is an uncertain and uncontrollable duration, a naive attempt at transforming the model would be to model observation itself as a contingent link (Figures 1a, 1b). This structure better equips us to reason about the controllability and execution of our original problem, but it is important to realize that this structure is not a valid STNU. In an STNU, all contingent links are required to start at an activated timepoint, whereas this transformation lets a contingent link start at a received timepoint. Nonetheless, while normal STNU algorithms may fail on this structure, we can still leverage it to better understand how to reason about variable-delay execution.

Under this new transformation, the next question we consider is how much information the observation of $\bar{\gamma}$ gives us in the worst case. Let $[a, b]$ be the bounds of the contingent link $X \Rightarrow E$. There are two ways to handle these contingent links based on how the uncertainty of the observation compares to the uncertainty of the original link.

Lemma 3. *If $b - a \le \bar{\gamma}^+(e) - \bar{\gamma}^-(e)$, we can express the same behavior in S' using $\gamma'(e) = \infty$.*

Proof. In this situation, there is at least as much uncertainty in the observation of the event as there is in the occurrence of the event, meaning we have no guarantee of receiving any
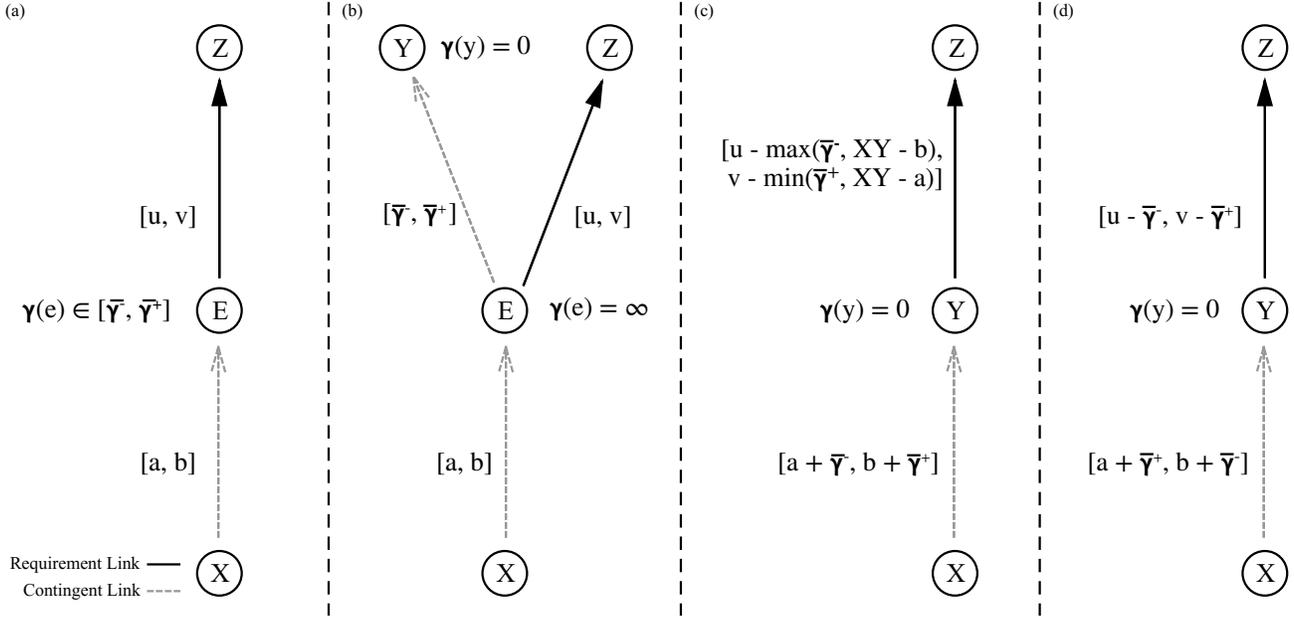
Figure 1: (a) A contingent link followed by a requirement link in our original STNU. (b) An equivalent (improper) STNU, which has a fixed-delay function instead of a variable-delay one. E becomes unobservable, and instead we immediately observe an explicit event Y after some uncertain delay. (c) An STNU that encodes a sufficient set of semantics to guarantee successful execution at runtime. XY refers to the true observed duration of the contingent link from X to Y. (d) A valid equivalent STNU, which has a fixed-delay function instead of a variable-delay one. The range of the contingent link shrinks, but the range of all attached requirement links must also shrink by a corresponding amount.

meaningful information after observing that the event happened. Consider the worst-case scenario where we learn that e happened $a + \bar{\gamma}^+(e)$ after the starting activated timepoint. It is clear that the original contingent link could have taken on a value of a given this information, but because $a + \bar{\gamma}^+(e) - b \geq \bar{\gamma}^-(e)$, we also know that the original link could have had a value of b and thus been anywhere in $[a, b]$. Since $\bar{\gamma}$ is not guaranteed to give us any information in this instance, in our derived STNU S', we can let $\gamma'(e) = \infty$.

We do not make controllability checking or network execution any harder by going from a highly uncertain observation to no observation at all. We always know the starting time of the original contingent link, as we choose it ourself, meaning we still have our original coarse bounds $[a, b]$ on when the event occurs. $\square$

Lemma 4. *If $b - a > \bar{\gamma}^+(e) - \bar{\gamma}^-(e)$, we can replace the bounds of the original link ending at e with $[a + \bar{\gamma}^+(e), b + \bar{\gamma}^-(e)]$.*

Proof. If $b - a > \bar{\gamma}^+(e) - \bar{\gamma}^-(e)$, we can take our original contingent link with range $[a, b]$ and variable-delay $\bar{\gamma}(e)$, and in S' we can transform it into a contingent link $[a + \bar{\gamma}^+(e), b + \bar{\gamma}^-(e)]$ with $\gamma'(e) = 0$ (see contingent link across Figures 1a, 1b, 1d), folding some of the uncertainty in observing e directly into the contingent link.

It is important to notice that the range of the modified contingent link is shorter than the range of possible times at which we would actually notice the occurrence of e, $[a + \bar{\gamma}^-(e), b + \bar{\gamma}^+(e)]$. Here again, we rely on the notion

that we are only interested in capturing the worst-case scenario. Imagine that we observed e at some time $a + \bar{\gamma}^+(e) - \epsilon$ where $0 < \epsilon \leq \bar{\gamma}^+(e)$. We know that the original link occurred at a lower-bound of a and that it has an upper-bound of $a + (\bar{\gamma}^+(e) - \bar{\gamma}^-(e)) - \epsilon$. In contrast, with an observation at $a + \bar{\gamma}^+(e)$, we have a strictly larger range of possible options for the original contingent link, meaning our restriction does not make the scheduling problem less constrained.

We can make the same argument for the upper-bound. If we observe e at some $b + \bar{\gamma}^-(e) + \epsilon$, then we have an upper-bound of b and a lower-bound of $b - (\bar{\gamma}^+(e) - \bar{\gamma}^-(e)) + \epsilon$. When we instead observe e at $b + \bar{\gamma}^-(e)$, we have the same upper-bound but a smaller lower-bound at $b - (\bar{\gamma}^+(e) - \bar{\gamma}^-(e))$, meaning the range of possible options is strictly larger. This means that our modified contingent link with $\gamma'(e) = 0$ fully captures the worst-case scenarios for e with our original S and $\bar{\gamma}$. $\square$

What remains is to demonstrate how to transform the requirement links attached to e such that they represent the original execution semantics of S. To validate that this is the case, we first examine what local execution semantics look like in a variable-delay temporal network (Figures 1b, 1c).

Lemma 5. *If we have contingent link $X \Rightarrow E$ with duration $[a, b]$, outgoing requirement link $E \rightarrow Z$ with duration $[u, v]$ with an unobservable E, and contingent link $E \Rightarrow Y$ with range $[\bar{\gamma}^-, \bar{\gamma}^+]$, we can replace the original requirement link during execution with a new link $Y \rightarrow Z$ with bounds $[u - max(\bar{\gamma}^-, XY - b), v - min(\bar{\gamma}^+, XY - a)]$, where XY is*

the true duration of $X \Rightarrow Y$. See Figure 1c for reference.

Proof. From an execution perspective, X and Y are the only events that can give us any information that we can use to reason about when to execute Z (since E is wholly unobservable).

If we execute Z based on what we learn from Y, then we use our information from Y to make inferences about the true durations of $X \Rightarrow E$ and $E \Rightarrow Y$ based on $X \Rightarrow Y$. We know that the lower-bound of $E \Rightarrow Y$ is at least $XY - b$ and that its upper-bound is at most $XY - a$. But we also have the a priori bounds on the contingent link that limit its range to $[\bar{\gamma}^-, \bar{\gamma}^+]$. Taken together, during execution we can infer that the true bounds of $E \Rightarrow Y$ are $[max(\bar{\gamma}^-, XY - b), min(\bar{\gamma}^+, XY - a)]$. Since we have bounds only on Z's execution in relation to E, we can then infer a requirement link $Y \to Z$ with bounds $[u - max(\bar{\gamma}^-, XY - b), v - min(\bar{\gamma}^-, XY - a)]$.

If we try to execute Z based on information we have from X, we must be robust to any possible value assigned to $X \Rightarrow E$. This means that we would be forced to draw a requirement link $X \to Z$ with bounds $[u + b, v + a]$. But we know that $u - max(\bar{\gamma}^-, XY - b) \le u + b - XY$ and $v - min(\bar{\gamma}^-, XY - a) \ge v + a - XY$, which means that the bounds we derived from Y are at least as expressive as the bounds that we would derive from X. $\square$

Since we have a local execution strategy that depends on the real value of XY, we can try to apply that strategy to the contingent link we restricted in Lemma 4 to repair the remaining requirement links.

Lemma 6. *If we have an outgoing requirement link $E \to Z$ with duration $[u, v]$ where E is a received timepoint, we can replace the bounds of the original requirement link with $[u - \bar{\gamma}^-, v - \bar{\gamma}^+]$. See Figure 1d for reference.*

Proof. If we directly apply the transformation from Figure 1c to our original STNU, we introduce a new complexity in the form of reasoning over min and max operations in our link bounds. However, from Lemma 4, we know that in a controllability evaluation context, it is acceptable for us to have the $X \Rightarrow Y$ link take on stricter range $[a + \bar{\gamma}^+, b + \bar{\gamma}^-]$ instead of $[a + \bar{\gamma}^-, b + \bar{\gamma}^+]$, meaning for the purposes of evaluating controllability, we can assume $a + \bar{\gamma}^+ \le XY \le b + \bar{\gamma}^-$. When we evaluate the requirement link $Y \to Z$, $max(\bar{\gamma}^-, XY - b) = \bar{\gamma}^-$ and $min(\bar{\gamma}^+, XY - a) = \bar{\gamma}^+$. This gives us the bounds for the requirement link that we see in Figure 1d. $\square$

Lemma 6 handles outgoing requirement edges connected to received timepoints, but we also must handle incoming edges.

Corollary 6.1. *If we have an incoming requirement link $Z \to E$ with duration $[u, v]$ where E is a received timepoint, we can replace the bounds of the original requirement link with $[u + \bar{\gamma}^+, v + \bar{\gamma}^-]$.*

Proof. A requirement link $Z \to E$ with bounds $[u, v]$ can be immediately rewritten as its reverse $E \to Z$ with bounds $[-v, -u]$. After reversing the edge, we can apply Lemma 6 to

Input: STNU S; variable-delay function $\bar{\gamma}$
Output: An STNU S' and fixed-delay function γ'
Initialization:

```
1  S' ← S.copy();
2  γ' ← {};
```

CONVERTTOFIXEDDELAY:

```
3   for l ∈ S'.contingentLinks() do
4       e ← l.endpoint();
5       a, b ← l.bounds();
6       if γ̄⁺(e) == ∞ or γ̄⁺(e) == γ̄⁻(e) then
7           γ'(e) ← γ̄⁺(e);
8       else if b − a ≤ γ̄⁺(e) − γ̄⁻(e) then
9           γ'(e) ← ∞;
10      else
11          e ← l.endpoint();
12          a, b ← l.bounds();
13          l.setBounds(a + γ̄⁺(e), b + γ̄⁻(e));
14          γ'(e) ← 0;
15          for l' ∈ e.outgoingReqLinks() do
16              u, v ← l'.bounds();
17              l'.setBounds(u − γ̄⁻(e), v − γ̄⁺(e));
18          for l' ∈ e.incomingReqLinks() do
19              u, v ← l'.bounds();
20              l'.setBounds(u + γ̄⁺(e), v + γ̄⁻(e));
21  return S', γ'
```

Algorithm 1: Algorithm for converting a variable-delay controllability problem to a fixed-delay controllability one.

get $Y \to Z$ with bounds $[-v - \bar{\gamma}^-, -u - \bar{\gamma}^+]$, which we can reverse again to get $Z \to Y$ with bounds $[u + \bar{\gamma}^+, v + \bar{\gamma}^-]$. $\square$

Now we put it all together and introduce Algorithm 1 as a complete algorithm for transforming an STNU S and variable-delay function $\bar{\gamma}$ into a corresponding STNU S' and fixed-delay function γ', where S is variable-delay controllable if and only if S' is fixed-delay controllable.

Theorem 7. *We can convert a variable-delay controllability problem into a corresponding fixed-delay controllability problem in $O(m + n)$.*

Proof. Initially in Algorithm 1 (lines 6-7), we handle Lemmas 1 and 2. Next, we check the condition set forth by Lemma 3 (lines 8-9) to see if in the worst-case the observation would fail to give us new information.

Finally, in the remaining part of the algorithm (lines 10-20), we transform all other contingent links and the requirement links that are associated with them. At line 13, we apply the transformation as specified by Lemma 4. At lines 15-20, we ensure that all outgoing and incoming requirement links are updated as specified in Lemma 6 and Corollary 6.1.

Overall, the transformation is efficient, running in linear time. Let n be the total number of timepoints and let m be the total number of constraints in an STNU. There are at most $O(n)$ contingent links, meaning the operations spanning lines 4-14 are applied at most $O(n)$ times. For each requirement link in S', we modify its bounds at most once as an outgoing link (lines 15-17) and once as an incoming link (lines 18-20).

This means that lines 15-20 are executed at most $O(m)$ times, meaning the total runtime is $O(m + n)$. $\qquad\square$

Since we can convert any variable-delay controllability problem into an equivalent-valued fixed-delay controllability problem, we can use fixed-delay controllability checkers to assess the variable-delay controllability of an STNU after transformation.

Theorem 8. *Variable-delay controllability can be evaluated in $O(n^3)$ time.*

Proof. Since the transformation takes $O(m+n)$ time without changing the size of the output STNU and fixed-delay controllability checking takes $O(n^3)$ [Bhargava *et al.*, 2018], the result is an $O(n^3)$ way to check variable-delay controllability for any STNU since $m \leq n^2$. $\qquad\square$

4 Variable-Delay Execution

Our algorithmic transformation gives us an efficient way to determine whether an STNU is variable-delay controllable but the resulting transformation does not give us an execution strategy for our STNU.

Intuitively, we want to use the resulting STNU from our fixed-delay transformation to guide execution but we face some limitations. To simplify our controllability checking, we restricted the ranges of many of our contingent links. If we tried to execute the resulting STNU, we would likely encounter a violation since it is possible for the world to violate the invariants of our STNU – namely that nature will respect the bounds of all contingent links. However, this problem is not insurmountable. We can use our transformed STNU as a guide for execution but amend our execution semantics slightly to account for these slight discrepancies.

Theorem 9. *Deriving an execution strategy for a variable-delay controllability problem reduces to finding an execution strategy for a fixed-delay controllability problem.*

Proof. The problem lies in a few contingent links that in the transformed STNU have bounds $[a + \bar{\gamma}^+, b + \bar{\gamma}^-]$ but in the original STNU have implicit bounds of $[a + \bar{\gamma}^-, b + \bar{\gamma}^+]$. The reason we were allowed to restrict these bounds is that the execution of other actions in the STNU were not dependent on the endpoint of this longer contingent link but rather on some other contingent link that had bounds $[a, b]$. In the restricted case, we always had less information about the true duration of the original contingent link, despite the fact that the range itself was smaller.

Armed with this knowledge, the remedy for our execution strategy is relatively straightforward. We follow the normal fixed-delay execution strategy for our derived STNU but with two exceptions. First, if the true duration of a contingent link is less than $a + \bar{\gamma}^+$, we buffer the response and act as if the duration was actually $a + \bar{\gamma}^+$. It is clear that waiting gives us no extra information.

Second, if more than $b + \bar{\gamma}^-$ time has passed and the contingent link has not reached completion, we act as if it actually reached completion at $b + \bar{\gamma}^-$. We can safely assume the earlier completion time because of the information it gives us. When $b + \bar{\gamma}^-$ time passes, the value of the original contingent link is somewhere in $[b - (\bar{\gamma}^+ - \bar{\gamma}^-), b]$. If we were to learn about the value at a later moment, $b + \bar{\gamma}^- + \epsilon$, then the value of the original contingent link would be in $[b + \epsilon - (\bar{\gamma}^+ - \bar{\gamma}^-), b]$, which is strictly tighter. Thus, if we assume an earlier completion time, we give ourselves a strictly harder problem, but we know it is still controllable because this still maps to our corresponding fixed-delay controllability problem.

These changes restrict the information we can learn about the original $[a, b]$ link, but since the system is still controllable with this restriction, our execution strategy remains valid. Thus, finding an execution strategy for an STNU with variable-delay function reduces to finding an execution strategy for an STNU with a fixed-delay function. $\qquad\square$

5 Empirical Evaluation

The introduction of variable-delay controllability gives us a level of expressiveness that we previously lacked. In this section, we attempt to characterize the gap in expressivity by showing how attempts to evaluate variable-delay problems using fixed-delay approximations lead to incorrect results.

To evaluate the comparative quality of the different approaches, we constructed a set of randomly generated STNUs. Each STNU had 10 contingent links with lower-bound 0 and an integer upper-bound uniformly chosen between 1 and 4. Each contingent link had a variable-delay function with a lower-bound of 0 and upper-bound chosen from the exponential distribution $f(t) = \lambda e^{-\lambda t}$ with $\lambda = 0.5$. For each pair of contingent link endpoints, we established a requirement link between them with probability $\frac{1}{40}$. Each requirement link had a lower-bound of 0 and an integer upper-bound uniformly chosen between 1 and 4. We chose these parameters because they represent a reasonable trade-off between simplicity in degenerate cases and sufficient complexity to exhibit interesting behaviors. Moving forward, we hope to validate our approach against real-world examples that demonstrate both temporal and observational uncertainty.

We employed three different strategies for our fixed-delay approximations: $\gamma(x_e) = \bar{\gamma}^-(x_e)$, $\gamma(x_e) = \frac{\bar{\gamma}^- + \bar{\gamma}^+}{2}$, and $\gamma(x_e) = \bar{\gamma}^+(x_e)$. For each strategy, we know that whenever the original STNU is variable-delay controllable with respect to $\bar{\gamma}$, it is also fixed-delay controllable with respect to γ. Each choice of γ represents a potential realization of the delays offered by $\bar{\gamma}$, and the fixed-delay approximation has the added benefit of eliminating uncertainty in observation.

We generated 1000 different STNUs and compared the variable-delay controllability results to the different fixed-delay controllability approaches (Table 1). The instances that are of greatest interest are those where the STNU is not variable-delay controllable but the fixed-delay approximations determines it to be controllable.

The false positive rate of the minimum fixed-delay controllability approximation is quite high at 39%, but the mean and maximum fixed-delay approximations have more reasonable false positive rates at 4.5% and 3.0% respectively. Since all approximations yield the correct answer when the original STNU is variable-delay controllable, it makes sense that the maximum fixed-delay approximation has the lowest false positive rate, as it is the most demanding of the three.

	Minimum fixed-delay controllable	Minimum fixed-delay uncontrollable	Mean fixed-delay controllable	Mean fixed-delay uncontrollable	Maximum fixed-delay controllable	Maximum fixed-delay uncontrollable
Variable-delay controllable	267	0	267	0	267	0
Variable-delay uncontrollable	289	444	33	700	21	712

Table 1: Variable-delay vs. minimum, mean, and maximum fixed-delay controllability results when using an exponential delay function with $\lambda = 0.5$.

We note that these results are also dependent on the width of the variable-delay ranges found in the network. We can simulate increasing the likelihood that a delay takes on a larger value by decreasing the choice of λ in our exponential delay function. When we vary our delay function using $\lambda = 0.5, 0.1, 0.05, 0.01$, the false positives of the max-delay approximation are 3.0%, 3.4%, 3.9%, 5.2%. As expected, this indicates that as the uncertainty of our bounds grows, there is an increasing advantage, from a correctness perspective, to using variable-delay controllability.

6 Conclusion

In this paper, we introduce variable-delay controllability as an extension to fixed-delay controllability over STNUs. We provide a formal definition showing how it generalizes fixed-delay controllability while also providing an efficient sound and complete algorithm for determining the variable-delay controllability of an STNU. Because variable-delay controllability execution reduces to fixed-delay controllability execution, we are able to demonstrate that variable-delay controllability is a formalism that can be used in practice to construct and evaluate schedules in the face of both temporal and observational uncertainty.

Acknowledgments

References

[Bhargava *et al.*, 2017] Nikhil Bhargava, Tiago Vaquero, and Brian Williams. Faster conflict generation for dynamic controllability. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-26, 2017*, pages 4280–4286, 2017.

[Bhargava *et al.*, 2018] Nikhil Bhargava, Christian Muise, Tiago Vaquero, and Brian Williams. Delay controllability: Multi-agent coordination under communication delay. In *DSpace@MIT*, 2018.

[Bit-Monnot *et al.*, 2016] Arthur Bit-Monnot, Malik Ghallab, and Félix Ingrand. Which contingent events to observe for the dynamic controllability of a plan. In *International Joint Conference on Artificial Intelligence (IJCAI-16)*, 2016.

[Casanova *et al.*, 2016] Guillaume Casanova, Cédric Pralet, Charles Lesire, and Thierry Vidal. Solving dynamic controllability problem of multi-agent plans with uncertainty using mixed integer linear programming. In *ECAI*, pages 930–938, 2016.

[Combi *et al.*, 2013] Carlo Combi, Luke Hunsberger, and Roberto Posenato. An algorithm for checking the dynamic controllability of a conditional simple temporal network with uncertainty. *Evaluation*, 1:1, 2013.

[Dechter *et al.*, 1991] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial intelligence*, 49(1-3):61–95, 1991.

[Fang *et al.*, 2014] Cheng Fang, Peng Yu, and Brian C Williams. Chance-constrained probabilistic simple temporal problems. 2014.

[Moffitt and Pollack, 2007] Michael D Moffitt and Martha E Pollack. Generalizing temporal controllability. In *IJCAI*, pages 1985–1990, 2007.

[Moffitt, 2007] Michael D Moffitt. On the partial observability of temporal uncertainty. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1031. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[Nilsson *et al.*, 2013] Mikael Nilsson, Jonas Kvarnström, and Patrick Doherty. Incremental dynamic controllability revisited. In *ICAPS*, 2013.

[Nilsson *et al.*, 2014] Mikael Nilsson, Jonas Kvarnström, and Patrick Doherty. Incremental dynamic controllability in cubic worst-case time. In *Temporal Representation and Reasoning (TIME), 2014 21st International Symposium on*, pages 17–26. IEEE, 2014.

[Shah *et al.*, 2007] Julie A Shah, John Stedl, Brian C Williams, and Paul Robertson. A fast incremental algorithm for maintaining dispatchability of partially controllable plans. In *ICAPS*, pages 296–303, 2007.

[Stedl and Williams, 2005] John Stedl and Brian C Williams. A fast incremental dynamic controllability algorithm. In *Proceedings of the ICAPS Workshop on Plan Execution: A Reality Check*, pages 69–75, 2005.

[Vidal and Fargier, 1999] Thierry Vidal and Helene Fargier. Handling contingency in temporal constraint networks: from consistency to controllabilities. *Journal of Experimental & Theoretical Artificial Intelligence*, 11(1):23–45, 1999.