Complexity Bounds for the Controllability of Temporal Networks with Conditions, Disjunctions, and Uncertainty (Extended Abstract)*

Nikhil Bhargava and **Brian C. Williams**
Massachusetts Institute of Technology
{nkb, williams}@mit.edu

Abstract

In temporal planning, many different temporal network formalisms are used to model real world situations. Each of these formalisms has different features which affect how easy it is to determine whether the underlying network of temporal constraints is consistent. While many of the simpler models have been well-studied from a computational complexity perspective, the algorithms developed for advanced models which combine features have very loose complexity bounds. In this work, we provide tight completeness bounds for strong, weak, and dynamic controllability checking of temporal networks that have conditions, disjunctions, and temporal uncertainty. Our work exposes some of the subtle differences between these different structures and, remarkably, establishes a guarantee that all of these problems are computable in PSPACE.

1 Introduction

In temporal planning, many different temporal formalisms are used to model real world situations. The choice of any particular type of network in modeling a problem has inherent trade-offs. If a temporal model supports more features, it can model a given scenario with higher fidelity. However, the additional features come at the expense of performance; modelers care about constructing schedules for temporal networks, and the presence of additional feature types can dramatically slow the runtime of scheduling algorithms.

The computational complexities of many of the simpler temporal models have been well-studied, but the same cannot be said of more advanced models. Despite this gap, there has been considerable effort put into constructing improved algorithms for these feature-rich temporal networks [Cimatti *et al.*, 2014; Cimatti *et al.*, 2016; Combi *et al.*, 2013; Hunsberger, 2016; Venable *et al.*, 2010].

The main contribution of this work is in providing significantly improved theoretical complexity bounds for computing the controllability of temporal networks with conditions, disjunctions, and temporal uncertainty. The existing bounds for some of these results have been quite loose with most decision problems not known to be better than EXPTIME and some not known to be better than EXPSPACE. We provide completeness results for the strong, weak, and dynamic controllability decision problems across these networks and remarkably prove that all of these problems can be solved in PSPACE. Our results are summarized in Figure 1. We conclude with a discussion of our results, giving practical advice to modelers who are interested in the trade-offs of using different temporal networks and lending insight into the differences between these networks.

2 Background

All temporal network formalisms consist of a series of timepoints corresponding to events and constraints dictating how these events relate to one another. The simplest variant of temporal networks is the Simple Temporal Network (STN) which requires constraints be binary relations that constrain the difference between two events [Dechter *et al.*, 1991]. In practice, modelers want additional features above and beyond what STNs offer to capture richer scenarios. We might want disjunctive constraints to indicate that we want to eat lunch either 30 minutes before swimming or immediately afterwards but not at any moment in between. Conditional constraints can be used to specify that taking an action is only required in certain situations. Temporal uncertainty is useful in modeling events that are outside of the scheduler's control, such as the effect of traffic on one's morning commute.

By adding these features in some combination, we get more expressive representations, but that expressivity comes at the cost of making schedule construction more difficult. In our work, we take advantage of prior work on Conditional Simple Temporal Networks (CSTNs) [Tsamardinos *et al.*, 2003], Temporal Constraint Satisfaction Problems (TCSPs) [Dechter *et al.*, 1991], Disjunctive Temporal Networks (DTNs) [Stergiou and Koubarakis, 2000], and Simple Temporal Networks with Uncertainty (STNUs) [Vidal and Fargier, 1999], and leverage those insights to prove new results about the computational complexity of Conditional Simple Temporal Networks with Uncertainty (CSTNUs) [Combi *et al.*, 2013], Temporal Constraint Satisfaction Problems with Uncertainty (TCSPUs) [Venable *et al.*, 2010], Disjunctive Temporal Networks with Uncertainty (DTNUs) [Venable

[*] This paper is an extended abstract of an article in the Artificial Intelligence Journal [Bhargava and Williams, 2019].

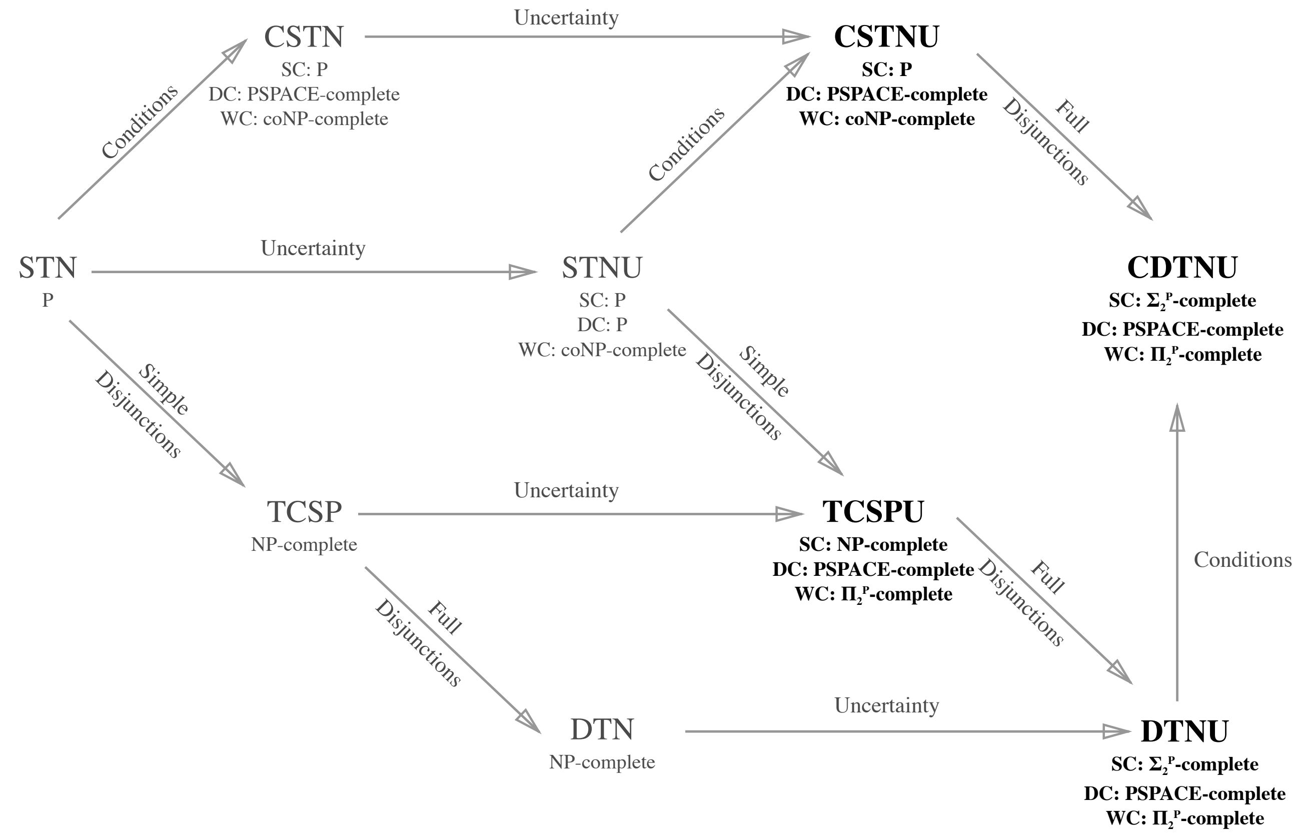

Figure 1: A taxonomic organization of temporal networks considered in this paper, how they relate to one another, and the complexity classes to which their decision problems belong. SC, DC, and WC represent strong controllability, dynamic controllability, and weak controllability, respectively. Results in bold represent novel results proved in our work.

and Yorke-Smith, 2005], and Conditional Disjunctive Temporal Networks with Uncertainty (CDTNUs) [Bhargava and Williams, 2019]. For brevity, we omit the formal definitions here and refer the reader to the respective works where these networks were introduced.

2.1 Polynomial Time Hierarchy

Before we continue to the actual complexity results it is useful to introduce the polynomial-time hierarchy [Stockmeyer, 1976], as it will allow us to more precisely characterize the difficulty of some of our controllability problems.

The classes Σ_k^P and Π_k^P are defined recursively. We start with $\Sigma_1^P = \mathrm{NP}$ and $\Pi_1^P = \mathrm{coNP}$ and define Σ_{k+1}^P as $\mathrm{NP}^{\Sigma_k^P}$ and Π_{k+1}^P as $\mathrm{coNP}^{\Sigma_k^P}$, where A^B represents the set of problems that can be solved in complexity class A if an oracle for a B-complete problem is provided.

In this paper, we will pay close attention to the complexity classes Σ_2^P and Π_2^P and will make heavy use of the fact that $\Sigma_k^P = \mathrm{co}\Pi_k^P$ and that $\forall\exists$3SAT is a Π_2^P-complete problem, where $\forall\exists$3SAT is the problem of determining whether for a given 3-CNF $\Phi(\vec{x}, \vec{y})$ it is the case that for all $\vec{y}$, there exists $\vec{x}$, such that $\Phi(\vec{x}, \vec{y})$ is true [Stockmeyer, 1976]. Σ_k^P and Π_k^P are also known to be fully contained within PSPACE, meaning that membership to any complexity class in the polynomial-time hierarchy guarantees the existence of a deterministic algorithm that uses at most polynomial space.

3 Complexity

While complexity results for the base temporal networks we have described are well-known, very few tight bounds exist for the networks derived from their composition, despite the fact that much work has been done to develop algorithms for them. Many of their hardness lower-bounds can be inherited from the base temporal networks, but it is an open question whether or not those bounds are tight.

In this section, we will provide complexity class completeness results for each of strong, weak, and dynamic controllability for each network, updating the hardness lower-bounds as needed before demonstrating membership to the appropriate class. When describing the controllability decision problems, we will use the prefixes SC-, WC-, and DC- to refer to checking the strong, weak, and dynamic controllability of the denoted temporal network, respectively.

3.1 Hardness Results

Our work starts by providing tighter hardness lower-bounds for the controllability problems across temporal networks. Existing results for CSTNs give us appropriate lower-bounds for CSTNUs, but for the temporal networks with disjunction and uncertainty, we need tighter analysis than the NP-hardness provided by TCSPs and DTNs.

Lemma 1. *Checking WC-TCSPU is Π_2^P-hard.*

Lemma 2. *Checking DC-TCSPU is PSPACE-hard.*

Lemma 3. *Checking SC-DTNU is Σ_2^P-hard.*

The corresponding proofs for the above lemmas can be seen in full in the original journal article [Bhargava and Williams, 2019], but in general the proofs follow by reduction. We use the canonical Π_2^P- and PSPACE-hard problems, ∀∃3SAT and TQBF and demonstrate how we can construct equivalent temporal networks that can be used to solve the same problems.

3.2 Completeness

With hardness results established, we now fully consider completeness for the controllability problems on each temporal network. Our general approach will be to map an inputted temporal network to a corresponding system of conditional linear inequalities that encode the same constraints. We will then use existential and universal quantifiers over the variables to dictate which type of controllability is being determined.

Our transformation proceeds as follows. We can imagine the execution of a temporal network as being a game played between two agents, the scheduler and nature, where the scheduler assigns times to executable timepoints and nature assigns times to contingent timepoints. In general the question of determining controllability will reduce to the problem of evaluating a quantified linear system and our techniques draw inspiration from and are related to approaches in those areas [Eirinakis *et al.*, 2014; Subramani, 2007].

For notational convenience, we will split our variables into $\vec{x}$ and $\vec{y}$ for those assigned by the scheduler and nature, respectively. For each executable timepoint e_i, we create a new variable x_i, and for each contingent timepoint c_i, we create a new variable y_i.

We create a one-to-one mapping between the set of temporal network constraints and the new linear inequalities. First, we replace all executable timepoints e_i with the corresponding x_i. With the contingent timepoints, however, we need to be more careful. For each contingent timepoint c_i, we find the contingent constraint that restricts it of the form $l_c \leq c_i - e_j \leq u_c$. We then replace each instance of c_i in our constraints with $y_i + x_j$. Our reason for doing this has to do with the nature of contingent constraints. In temporal networks, there is a guarantee that nature respects the contingent constraint bounds in relation to its corresponding starting executable timepoint. So while free constraints relate timepoints in terms of the absolute time of their occurrence, contingent constraints require nature to respect relative timings of events. If the durations of contingent constraints are to be known before scheduling begins, as is the case with weak controllability, then the constructed system of linear inequalities will fail to map to the base temporal network if nature is asked to pick the precise times of contingent events.

After the substitutions, each constraint is a combination of conditional linear inequalities of the form $\psi \rightarrow \vec{a} \cdot \begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} \leq b$, where b is some constant, ψ is a (possibly empty) precondition for the enforcement of the constraint, and $\vec{a}$ represents the coefficients of the constraints where each coefficient is either -1, 0, or 1. Since all constraints are relative, without loss of generality, we can say that the earliest event happens at time $t = 0$, meaning we can safely require that $\vec{x} \geq 0$. When we quantify over variables to pick controllability, we require that each x_i has an existential quantifier and each y_i has a for-all quantifier drawn from the union of the ranges $[l_1, u_1], ..., [l_d, u_d]$, where l_j and u_j are retrieved from one of c_i's corresponding contingent constraints.

When evaluating controllability for disjunctive networks, it is useful to consider each contingent range separately, and so we will define Ω as a mapping from each variable y_i and one of its possible continuous ranges. In general, we will use the shorthand $\forall\Omega$ to indicate that we are considering all possible mappings and $\forall \vec{y} \in \Omega$ to specify that we are drawing our $\vec{y}$ from one particular mapping. Our choice of the ordering of the quantifiers will dictate which type of controllability will be considered. We also must consider how conditions affect our model, and will define Ψ as the full set of conditions that can be observed by the scheduler when our temporal networks include conditional constraints.

As with the lemmas in the previous subsection, the full proofs for the below theorems can be found in [Bhargava and Williams, 2019].

Theorem 4. *Checking SC-CSTNU is in P.*

Theorem 5. *Checking SC-TCSPU is NP-complete.*

Theorem 6. *Checking SC-DTNU and SC-CDTNU are Σ_2^P-complete.*

Theorem 7. *Checking WC-CSTNU is coNP-complete.*

Theorem 8. *Checking WC-TCSPU, WC-DTNU, and WC-CDTNU are Π_2^P-complete.*

Our approach for providing a bound on the runtime of dynamic controllability across our networks is slightly different. Unlike the previous approaches, we do not have a natural way to translate a temporal network to a quantified linear program because the order in which values are assigned to variables is not deterministic. Instead, we introduce Algorithm 1 as a procedure for checking the dynamic controllability of a CDTNU. This algorithm can be used for checking the dynamic controllability of all temporal networks that we have described.

Theorem 9. *Checking DC-CSTNU, DC-TCSPU, DC-DTNU, and DC-CDTNU are all PSPACE-complete.*

The formal proof of Theorem 9 can be found in [Bhargava and Williams, 2019], but it follows directly from Algorithm 1. Our algorithm recursively enumerates all possible assignments and situations that can be encountered during temporal network execution and returns true if it is always possible to reactively construct a valid schedule. Our strategy is careful

Algorithm 1 CHECKDC: PSPACE algorithm for checking DC-CDTNU.

Input: A list of timepoints with assigned values, T
A list of active contingent links, A
A set of yet-to-be-executed timepoints E
The input CDTNU G
The current time, τ **Output**: Whether the CDTNU is dynamically controllable.

```
if $E.empty()$ then
    for $realization \in A.realizationsFrom(\tau)$ do
        if !$G.isConsistent(T.extend(realization))$ then
            return $false$
    return $true$
for $t \in E$ do
    for $\tau' \in [\tau, G.tMax]$ do
        $allSatisfied \leftarrow true$
        for $realization \in A.realizationsFrom(\tau)$ do
            $earliest \leftarrow realization.earliest()$
            if $earliest.time \leq \tau'$ then
                if !CHECKDC($T \cup \{earliest\}$,
                $A.nextContingents(earliest)$,
                $E$,
                $G$,
                $earliest.time$) then
                    $allSatisfied \leftarrow false$
                    break
            else
                if !CHECKDC($T \cup \{$ASSIGN$(t, \tau')\}$,
                $E$,
                $G$,
                $earliest.time$) then
                    $allSatisfied \leftarrow false$
                    break
        if $allSatisfied$ then
            return $true$
```

to only consider at most n recursive calls at a time and uses a polynomial amount of space per invocation, meaning the entire procedure runs in PSPACE.

4 Discussion and Conclusions

Our work provides novel complexity results that are much tighter than existing bounds and require at most polynomial space for strong, weak, and dynamic controllability of several distinct types of temporal networks; this work is summarized in Figure 1. Beyond the contribution of the relevant proofs, the value of these results is that it gives modelers insight into which types of features have a significant impact on the runtime complexity of a problem. Many of these insights are not immediately obvious, and in the remainder of this section we discuss a few of them.

First we consider CSTNUs. CSTNUs are a generalization of CSTNs and STNUs and share much in common with their predecessors. In particular, strong controllability of CSTNUs, being in P, can be computed quite efficiently. Our work actually proves a stronger result that a CSTNU is strongly controllable if and only if the corresponding STNU derived by making all constraints unconditional is strongly controllable. This implies that strong controllability of CSTNUs can be computed in $O(mn)$ time, which is as fast as it takes to compute the feasibility of a simple STN. When we turn to weak and dynamic controllability, we see that checking the controllability of a CSTNU is in the same class as checking controllability of a CSTN. From the perspective of the modeler, this implies that there is a surprisingly low cost to adding uncertainty to a temporal constraint model that already uses conditional constraints.

While CSTNU controllability checking matches the complexity of CSTN controllability checking, it only matches the controllability checking complexity of strong and weak controllability for STNUs. In fact, dynamic controllability checking across all types of networks, with the exception of STNUs, is PSPACE-complete. In scheduling problems, modelers must often make the trade-off between using strong controllability, which is often easier to compute, and dynamic controllability, which gives more flexibility during execution but is more expensive. In instances where dynamic controllability is deemed necessary, there is a significant advantage to relaxing the underlying temporal model, eliminating conditional and disjunctive constraints, to use an STNU. It is still quite surprising that despite the fact that STNU dynamic controllability can be determined in polynomial time, every other modification makes computing dynamic controllability PSPACE-complete.

A final area worth discussing is the effect of temporal disjunctions. The two temporal network models that use disjunctions without temporal uncertainty are TCSPs and DTNs; TCSPs have simple disjunctions, only requiring disjunctions over a single link, while DTNs have full disjunctions, allowing disjunctions to span multiple links. Since determining feasibility for both network structures is NP-complete, intuition would suggest that after adding uncertainty the complexity of checking controllability for TCSPUs and DTNUs would also be the same. While this is the case for weak and dynamic controllability, we do see a difference in strong controllability, meaning that strong controllability is easier to compute in TCSPUs than it is in DTNUs, assuming NP $\neq \Sigma_2^P$, implying there is a meaningful difference between the two types of disjunctions.

As we look forward, there are still many areas worthy of future research efforts. One, in particular, is the development of novel algorithms for determining the controllability of these networks. Our work establishes bounds on the complexity of computing controllability but does minimal work to provide algorithms for doing so. Our proofs admit the trivial polynomial-space strategy of recursive enumeration of certificates but this approach is likely impractical. Our results open the challenge of finding novel algorithms that are reasonable for practical use while still respecting polynomial bounds.

References

[Bhargava and Williams, 2019] Nikhil Bhargava and Brian C Williams. Complexity bounds for the controllability of temporal networks with conditions, disjunctions, and uncertainty. *Artificial Intelligence*, 271:1–17, 2019.

[Cimatti *et al.*, 2014] Alessandro Cimatti, Luke Hunsberger, Andrea Micheli, Roberto Posenato, and Marco Roveri. Sound and complete algorithms for checking the dynamic controllability of temporal networks with uncertainty, disjunction and observation. In *The 21st International Symposium onTemporal Representation and Reasoning (TIME)*, pages 27–36. IEEE, 2014.

[Cimatti *et al.*, 2016] Alessandro Cimatti, Andrea Micheli, and Marco Roveri. Dynamic controllability of disjunctive temporal networks: Validation and synthesis of executable strategies. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 3116–3122, 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. In *The 5th International Conference on Agents and Artificial Intelligence*, volume 1, page 1, 2013.

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

[Eirinakis *et al.*, 2014] Pavlos Eirinakis, Salvatore Ruggieri, K Subramani, and Piotr Wojciechowski. On quantified linear implications. *Annals of Mathematics and Artificial Intelligence*, 71(4):301–325, 2014.

[Hunsberger, 2016] Luke Hunsberger. Efficient execution of dynamically controllable simple temporal networks with uncertainty. *Acta Informatica*, 53(2):89–147, 2016.

[Stergiou and Koubarakis, 2000] Kostas Stergiou and Manolis Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence*, 120(1):81–117, 2000.

[Stockmeyer, 1976] Larry J Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[Subramani, 2007] K Subramani. On a decision procedure for quantified linear programs. *Annals of Mathematics and Artificial Intelligence*, 51(1):55–77, 2007.

[Tsamardinos *et al.*, 2003] Ioannis Tsamardinos, Thierry Vidal, and Martha E Pollack. Ctp: A new constraint-based formalism for conditional, temporal planning. *Constraints*, 8(4):365–388, 2003.

[Venable and Yorke-Smith, 2005] Kristen Brent Venable and Neil Yorke-Smith. Disjunctive temporal planning with uncertainty. In *International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1721–1722, 2005.

[Venable *et al.*, 2010] Kristen Brent Venable, Michele Volpato, Bart Peintner, and Neil Yorke-Smith. Weak and dynamic controllability of temporal problems with disjunctions and uncertainty. In *Workshop on Constraint Satisfaction Techniques for Planning & Scheduling*, pages 50–59, 2010.

[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.