SMT-based Repair of Disjunctive Temporal Networks with Uncertainty: Strong and Weak Controllability

Ajdin Sumic[1], Alessandro Cimatti[2], Andrea Micheli[2], and Thierry Vidal[1]

[1] LGP/ENIT, Technical University of Tarbes, France
[2] Fondazione Bruno Kessler, Italy.

Abstract. Temporal Networks with Uncertainty are a powerful and widely used formalism for representing and reasoning over temporal constraints in the presence of uncertainty. Since their introduction, they have been used in planning and scheduling applications to model situations where some activity durations or event timings are not under the control of the scheduling agent. Moreover, a wide variety of classes of temporal networks (depending on the types of constraints) have been defined, and many algorithms for dealing with these emerged.
We are interested in the repair problem, consisting of reconsidering the bounds of the uncertain durations when the network is not *controllable*, i.e., when no strategy for scheduling exists. This problem is important, as it allows for the explanation and negotiation of the problematic uncertainties. In this paper, we address the repair problem for a very expressive class of temporal networks, namely the Disjunctive Temporal Networks with Uncertainty. We use the Satisfiability Modulo Theory framework to formally encode and solve the problem, and we devise a uniform solution encompassing different "levels" of controllability, namely strong and weak. Moreover, we provide specialized encodings for important subclasses of the problem, and we experimentally evaluate our approaches.

Keywords: Disjunctive Temporal Network with Uncertainty · Plan Repair · Satisfiability Modulo Theory · Strong and Weak Controllability.

1 Introduction

Since their introduction in [11], Temporal Networks have been recognized as a fundamental tool to represent and reason about temporal dependencies among tasks and events. Several variants have been studied in depth: e.g. Simple, Disjunctive [21, 4]. The important extension to temporal networks with uncertainty [25] allows to model tasks of uncontrollable duration. This yields the problem of controllability, i.e., devising a strategy that will work regardless of the uncertainty, that comes in different forms depending on the observability assumptions, i.e., Weak, Strong, and Dynamic controllability. Various algorithms have been proposed to *check* controllability, also based on constraints propagation[25, 17], Satisfiability Modulo Theories [3], and Timed Games[8].

Despite the widespread use of these formalisms in the literature, little attention has been devoted to the case in which a temporal network is deemed non-controllable.

In this case, the source of non-controllability could be unfeasible requirements (e.g., even with a less uncertain environment, it would still be impossible to schedule the controllable time points). Still, the assumptions about the environment could also be too loose given the constraints (i.e., if only an uncontrollable activity had stricter bounds, we could recover controllability). In this paper, we tackle the problem of *repair* of a Disjunctive Temporal Network with Uncertainty (DTNU) [24]. Starting from an uncontrollable network $\mathcal{D}$, we define the repair problem as finding a variant $\mathcal{D}'$ that is controllable and that is "sufficiently" close to $\mathcal{D}$. In particular, we consider the case where the $\mathcal{D}'$ variant is obtained from $\mathcal{D}$ by restricting the bounds on uncontrollable durations.

This problem is of theoretical importance but also has a strong practical relevance: consider the case of a multi-agent system, in which the uncontrollability bounds can be negotiated, for example, when the duration of an activity is considered to be uncontrollable for the scheduling agent, but it is, in fact, controllable for another agent. In a collaborative scenario, the agent can try to repair its own network by asking for stronger assumptions on the controlling agent instead of relaxing its objectives by creating a new controllable network. In other words, one agent could ask the controlling agent to reduce its flexibility in order to help in recovering controllability. Another application of the repair problem concerns explainability: by providing a (minimal) repair to an uncontrollable network, we suggest a strategy to change the constraints to recover controllability: this is an immediate counterfactual explanation [22] for the non-controllability and, even if the constraint is not negotiable, explanation allows to focus on where/what to replan.

We tackle the repair problem in the setting of Satisfiability Modulo Theories, proposing a solution for the cases of Weak and Strong controllability. We first define a general approach for DTNUs that uniformly works for both controllability levels: we define a quantified formula encoding all valid repairs for a given DTNU. Any model of such formula is a valid relaxation of the contingent bounds that recovers the controllability of the network. Furthermore, we prove that if the formula is unsatisfiable, then the problem admits no repair. Then, we use Optimization Modulo Theory [20] to select, among the possible repairs the one that sacrifices the least amount of flexibility. Second, we devise a specialized encoding for the specific but important case of Simple Temporal Networks with Uncertainty (STNU) (that is, when disjunctions are not allowed). By exploiting the convexity of the problem formulation, we obtain a much more efficient algorithm for synthesizing a repaired STNU.

We implemented the proposed approaches within the pySMT framework [12], using the Z3 solver as backend [10], and we experimentally evaluated them on a large set of non-controllable DTNU and STNU benchmarks. The empirical evaluation shows that the general case of DTNU repair may incur scalability issues, heavily dependent on the number of uncontrollables. On the other hand, the specialized algorithm leads to substantial increases in efficiency by leveraging the features of STNUs, hence reducing the cost of quantified reasoning.

Structure of the paper The next Section discusses the related work. In Section 3, we present the needed background and in Section 4 we define the problem of DTNU repair. In Section 5, we propose a general repair encoding for the general case of DTNU, while Section 6 tackles the special case of STNU. In Section 7, we empirically evaluate our approaches and in Section 8 we draw our conclusions and present directions for future work.

2 Related Work

The problem of repair for Disjunctive Temporal Networks under Uncertainty has never been tackled before. Most of the literature considered problems of checking controllability [16][13][15] and flexibility in execution [19]. The closest related works typically focus on STNUs and diagnosis, i.e., pinpointing reasons for non-controllability. For example, in [14], a diagnosis approach for the temporal checker is proposed for dynamic controllability. This approach is able to give the set of constraints involved in the non-DC of an STNU. Then, the *designer*(planner) is in charge of solving the problem. In the case where multiple contingents are involved, the repair will help the designer in deciding what to do by providing the best way to shrink the contingent to restore controllability.

The notion of repair arises in [6]. The authors propose a MILP approach to decouple an STNU into sub-networks (STNU), one per agent, in a multi-agent setting. The authors state that it should be possible to modify the encoding of the MILP approach so that it reduces the bounds of the contingents so that all sub-networks are DC. Then, in [2], the authors compute the volume space of an STNU to assess just how far from being controllable an uncontrollable STNU is by defining some metrics for SC and DC. Later, they propose an incomplete LP approach to repair a non-DC STNU by repairing the negative cycles [1]. However, it is incomplete because it doesn't consider inter-dependence between negative cycles.

In this paper, we formally define the repair problem as tightening the contingent constraints and propose a series of SMT encodings for synthesizing valid repairs. Unlike the literature reported above, we consider the very general case of DTNUs (instead of limiting ourselves to STNUs) and uniformly tackle both strong and weak controllability. Moreover, we provide specialized encodings for the STNU case.

3 Background

3.1 Satisfiability and Optimization Modulo Theory

The Satisfiability Modulo Theory (SMT) [3] is the problem of deciding whether there exists a model that satisfies a first-order formula ϕ (i.e., an assignment to the free variables in ϕ such that ϕ is satisfied. Given the formula $\phi_1 = (x <= y) \wedge (x + y = 10)$ a valid model of ϕ_1 is $\{x := 4,\ y := 6\}$, where $x, y \in \mathbb{R}$.

SMT solvers can support multiple *Theories* such as Linear Real Arithmetic (LRA). In LRA a formula is a Boolean combination, or a universal and existential quantification (respectively $\forall$, $\exists$) of atoms, which are linear constraints in the form: $\sum_i a_i x_i \bowtie c$ where $\bowtie \in \{>, <, \leqslant, \geqslant, \neq, =\}$, every x_i is a real variable and every a_i, c are real constants. *QF_LRA* denotes the quantifier-free fragment of LRA.

Optimization Modulo Theory (OMT) generalizes SMT with optimization procedures to find a model of ϕ that is optimal for an objective function f (or a combination of multiple objective functions) under all models of a formula ϕ. The objective function f can be expressed as a term in different theories, but this paper focuses only on objective functions expressed as *QF_LRA* terms. In the literature, several OMT solvers exist, such as OptiMathSAT [20] and Z3 [5].

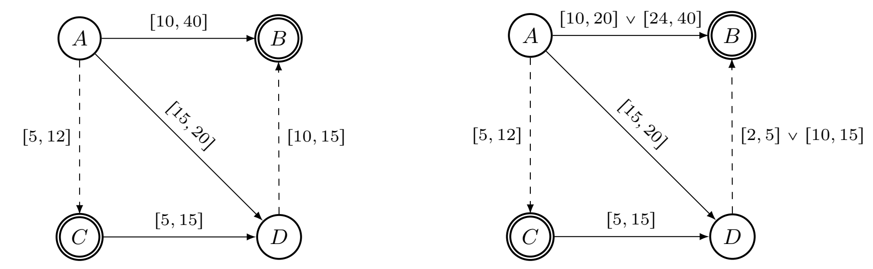
3.2 Temporal Networks (With Uncertainty)

A Temporal Network (TN) is a formalism that is used to represent temporal constraints over time-valued variables called time points. Two families of TN exist in the literature: TN that does not consider uncertainty, to which the scheduler freely assigns all time points [11, 23], and TN with Uncertainty (TNU), where only a subset of time points are assigned by the scheduler, while another exogenous entity decides the others [25, 24]. This paper focuses on the Disjunctive Temporal Network with Uncertainty (DTNU) and Simple Temporal Network with Uncertainty (STNU), which is a restricted case of DTNU.

Definition 1. *A **DTNU** is a tuple $\langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$ where:*

- *$\mathcal{V}$ is a set of time points, partitioned into controllable ($\mathcal{V}_c$) and uncontrollable ($\mathcal{V}_u$);*
- *$\mathcal{E}$ is a set of free constraints (requirement constraints): each constraint e_i is of the form, $\bigvee_{j=1}^{D_i} v_{1j} - v_{2j} \in [L_{ij}, U_{ij}]$, for some $v_{1j}, v_{2j} \in \mathcal{V}$ and $L_{ij}, U_{ij} \in \mathbb{R} \cup \{+\infty, -\infty\}$;*
- *$\mathcal{C}$ is a set of contingent constraints: each $c_i \in \mathcal{C}$ is of the form, $\langle b_i, \mathcal{B}_i, d_i \rangle$, where $b_i \in \mathcal{V}_c$, $d_i \in \mathcal{V}_u$, and $\mathcal{B}_i$ is a set of pairs $\langle L_{ij}, U_{ij} \rangle$ such that $0 \leqslant L_{ij} \leqslant U_{ij} < \infty$, $j \in [1, |\mathcal{B}_i|]$; and for any distinct pairs, $\langle L_{ij}, U_{ij} \rangle$ and $\langle L_{ik}, U_{ik} \rangle \in \mathcal{B}_i$, either $L_{ij} > U_{ik}$ or $U_{ij} < L_{ik}$.*

In a DTNU, free constraints are disjunctions of intervals, each between any time points, possibly overlapping, while a contingent is a disjunction of non-overlapping intervals restricted to a single pair of time points.
Intuitively, controllable time points ($\mathcal{V}_c$) are moments in time to be decided by the scheduling agent, which is trying to satisfy all the free constraints ($\mathcal{E}$) under any possible instantiation of the uncontrollable time points ($\mathcal{V}_u$) respecting the contingent constraints ($\mathcal{C}$).



(a)

$\vec{x} = \{A, D\} \qquad \vec{y} = \{y_C, y_B\}$

$\Gamma(\vec{y}) = (y_C \geqslant 5) \wedge (y_C \leqslant 12) \wedge (y_B \geqslant 10) \wedge (y_B \leqslant 15)$

$\Psi(\vec{x}, \vec{y}) = ((D + y_B) - A \geqslant 10) \wedge ((D + y_B) - A \leqslant 40) \wedge$
$((y_C + D) - A \geqslant 15) \wedge$
$((y_C + D) - A \leqslant 20) \wedge (D - y_C \geqslant 5) \wedge (D - y_C \leqslant 15)$

(b)

$\vec{x} = \{A, D\} \qquad \vec{y} = \{y_C, y_B\}$

$\Gamma(\vec{y}) = (y_C \geqslant 5) \wedge (y_C \leqslant 12) \wedge$
$(((y_B \geqslant 2) \wedge (y_B \leqslant 5)) \vee (y_B \geqslant 10) \wedge (y_B \leqslant 15)))$

$\Psi(\vec{x}, \vec{y}) = (D - y_C \geqslant 5) \wedge (D - y_C \leqslant 15) \wedge$
$((y_C + D) - A \geqslant 15) \wedge ((y_C + D) - A \leqslant 20) \wedge$
$(((D + y_B) - A \geqslant 10) \wedge ((D + y_B) - A \leqslant 20)) \vee$
$((D + y_B) - A \geqslant 24) \wedge ((D + y_B) - A \leqslant 40)))$

Fig. 1: Graph representations of the STNU (a) and DTNU (b) examples, together with their SMT encodings. Nodes represent time points, doubly circled nodes are uncontrollable; solid edges are free constraints while dashed edges are contingent constraints.

Definition 2. *An **STNU** is a DTNU without disjunctions. Hence, an STNU is a DTNU $\langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$ where $D_i = 1$ for all free constraints and $|\mathcal{B}_i| = 1$ for all contingent constraints.*

Example 1. Bob is working in a production line and has 40 minutes to build a product. This product is done in 3 phases: first Bob needs to wait for a machine to transform raw materials into a component of the final product. This task performed by the machine has an uncontrollable duration that lasts between 5 to 12 minutes. Then, Bob needs to compare the quality of the component with some regulation (size, weight, etc.) and he can decide to take 5 to 15 minutes to do the checking. However, Bob knows that to be efficient the two tasks need to be performed between 15 to 20 minutes. Then, he must give the component to another machine that will create the final product. This task is an uncontrollable event that lasts between 10 to 15 minutes. An STNU representing the example is shown in Figure 1a. For the sake of the example, we suppose that the product needs at least some minimal amount of time to be produced; therefore we put a minimum duration of 10. A DTNU version is shown in Figure 1b, where Bob can produce two products using two different machines: the first product that needs to be produced in 20 minutes with the first robot that can produce it in 2 to 5 minutes, the second needs to be produced between 24 to 40 minutes with the second machine that can produce it in 10 to 15 minutes. We will explain the encoding of the two examples in Section 3.3.

The basic computational problem arising given a DTNU is controllability [25]: we want to check whether there exists a strategy for scheduling the controllable time points that will respect all the free constraints under any possible realization of the uncontrollable time points respecting the contingent constraints. However, depending on the assumptions on the observability of uncontrollable time points, different *levels* of controllability have been defined. In this paper, we focus on strong and weak controllability, corresponding to the cases of no-observation and clairvoyant observation, respectively. In the following, we report the basic definitions for these two cases.

Definition 3. *A* ***controllable assignment*** $\delta : \mathcal{V}_c \rightarrow \mathbb{R}$ *of a DTNU* $\langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$ *is a mapping from all the controllable time points to real values.*

Definition 4. *A* ***DTN*** $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \varnothing \rangle$ *is a DTNU with no contingent constraints (and no uncontrollable time points:* $\mathcal{V}_c = \mathcal{V}$, $\mathcal{V}_u = \varnothing$*). A* ***consistent schedule*** *for a DTN* $\mathcal{D}$ *is a controllable assignment* δ *that satisfies each free constraint* e_i *in* $\mathcal{E}$*:*

$$\bigvee_{j=1}^{D_i} \delta(v_{1j}) - \delta(v_{2j}) \in [L_{ij}, U_{ij}].$$

Definition 5. *given a DTNU* $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$*, let* $m = |\mathcal{V}_u|$*. The* ***situations*** *of* $\mathcal{D}$ *is a set of tuples* $\Omega_{\mathcal{D}}$ *defined as the cartesian product of:*

$$\bigtimes_{\langle b_i, \mathcal{B}_i, d_i \rangle \in \mathcal{C}} \left(\bigcup_{\langle L_{ij}, U_{ij} \rangle \in \mathcal{B}_i} [L_{ij}, U_{ij}] \right).$$

A ***situation*** *is an element* ω *of* $\Omega_{\mathcal{D}}$ *and we write* $\omega(c)$ *with* $c \in \mathcal{C}$ *to indicate the element in* ω *associated with* c *in the cross product.*

Intuitively, the set of situations defines the region of uncertainty, all the allowed durations of contingent constraints. A network will be deemed controllable if (with the provided observations) it is possible to schedule the controllable time points so that all free constraints are satisfied in any possible situation.

Definition 6. *Given a DTNU* $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$ *and a situation* ω*, the* ***projection*** $\mathcal{D}_\omega$ *is the DTN* $\langle \mathcal{V}, \mathcal{E} \cup \mathcal{C}_\omega, \varnothing \rangle$ *where* $\mathcal{C}_\omega = \{d_i - b_i \in [\omega(c), \omega(c)] \mid c \in \mathcal{C}$ *and* $c = \langle b_i, \mathcal{B}_i, d_i \rangle\}$.

Intuitively, the projection $\mathcal{D}_\omega$ substitutes all the contingent links with requirements, forcing the duration of a contingent link to the value indicated in ω.

Definition 7. *A DTNU* $\mathcal{D}$ *is* ***weakly controllable*** *iff for all* $\omega \in \Omega$*, there exists a consistent schedule* δ *for* $\mathcal{D}_\omega$*.*

In weak controllability, we assume that the uncontrollable durations are decided before execution, and therefore, the scheduling agent can condition its decisions on the situation.

Definition 8. *A DTNU $\mathcal{D}$ is* ***strongly controllable*** *iff there exists a consistent schedule δ, for all $\mathcal{D}_\omega$ where $\omega \in \Omega$.*

This level of controllability implies that the controllable schedule is so robust that it's possible to fix the execution of the controllable time points in time so that it carries any execution of the uncontrollable time point observed at execution time.

A third level of controllability exists, dynamic controllability, but it's not in the scope of this paper. However, many works in the literature tackled this level of controllability, which is of great practical importance [16, 13, 17].

3.3 Controllability Checking using SMT

In this section, we present the basic SMT encodings devised in [7] and [9], targeting strong and weak controllability, respectively. We limit ourselves to the basic formulations, as these are the only requirements for our paper. Still, the authors provide different variations of these encodings that turn out to be more efficient than the basic formulation for the sake of checking controllability.

Definition 9. *Given a DTNU $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$, we define the following sets of SMT variables:*

- *the* ***uncontrollable SMT durations*** *$\vec{y}$ is a set of real SMT variables, one for each contingent constraint $c_i \in \mathcal{C}$ (we write y_i to indicate the variable corresponding to c_i);*
- *the* ***controllable SMT timepoints*** *$\vec{x}$ is a set of real SMT variables, one for each controllable time point (we write x_v to indicate the variable corresponding to $v \in \mathcal{V}_c$).*

We also define the following basic SMT formulae:

- *$\Gamma(\vec{y})$ representing the* ***SMT contingent constraints****:*

$$\Gamma(\vec{y}) \doteq \bigwedge_{c_i \doteq \langle b_i, \mathcal{B}_i, d_i \rangle \in \mathcal{C}} \left(\bigvee_{\langle L_{ij}, U_{ij} \rangle \in \mathcal{B}_i} (y_i \geqslant L_{ij} \wedge y_i \leqslant U_{ij}) \right) \tag{1}$$

- *$\Psi(\vec{x}, \vec{y})$ representing the* ***SMT free constraints*** *(requirements):*

$$\Psi(\vec{x}, \vec{y}) \doteq \bigwedge_{e_i \in \mathcal{E}} (((e_i[(b_1 + y_1)/d_1])[(b_2 + y_2)/d_2]) \dots [(b_m + y_m)/d_m]) \tag{2}$$

where $m = |\mathcal{C}|$ and the logical notiation $\phi[a/b]$ indicates the substitution of the term b with term a in the formula ϕ.

In this approach, each uncontrollable time point $d_i \in \mathcal{V}_u$ is encoded by its difference with the starting time point $b_i \in \mathcal{V}_c$ represented by a universally quantified variable $y_i \in \mathbb{R}$ such that: $d_i - b_i \in [L_{ij}, U_{ij}]$, and $y_i \in [L_{ij}, U_{ij}]$. Thus, y_i represents the duration of the interval $[b_i, d_i]$ and the execution of d_i is $(b_i + y_i)$. We define $\vec{x}$, $\vec{y}$ as the sets of controllable time points and uncontrollable durations, respectively. Therefore, the rewriting of the contingent constraints depends only on $\vec{y}$. Next, $\Gamma(\vec{y})$ is the formula representing the conjunction of all the contingent constraints over the uncontrollable durations, and $\Phi(\vec{x}, \vec{y})$ is the formula representing the conjunction of all free constraints over $\vec{x}$ and $\vec{y}$. We show in Figures 1a and 1b this encoding for the running examples.

Given the formulae presented in Definition 9, the authors of [7] prove that a DTNU $\mathcal{D}$ is weakly controllable if the following formula Φ_{weak} is valid.

$$\Phi_{weak} \doteq \forall \vec{y}. \exists \vec{x}. \Gamma(\vec{y}) \rightarrow \Psi(\vec{x}, \vec{y}) \tag{3}$$

Moreover, in [9], they prove that $\mathcal{D}$ is strongly controllable if the following formula ϕ_{strong} is valid.

$$\Phi_{strong} \doteq \exists \vec{x}. \forall \vec{y}. \Gamma(\vec{y}) \rightarrow \Psi(\vec{x}, \vec{y}) \tag{4}$$

We will use these encodings as the basis for our repair problem formulation.

4 DTNU Repair: Problem Definition

The concept of repair in DTNU arises when the network is not controllable; in these situations, we want to find a tightening (if it exists) of the bounds of the contingent constraints such that controllability is recovered. This is useful as a counterfactual explanation for non-controllability and in multi-agent applications where contingents are controlled by other agents and can be negotiated. In the following, we formalize the repair problem.

Definition 10. *Let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$ be a not τ-controllable DTNU, with $\tau = \{weak, strong\}$. The τ-**repair problem** consists in finding a τ-controllable DTNU $\mathcal{D}' = \langle \mathcal{V}, \mathcal{E}, \mathcal{C}' \rangle$ such that $\mathcal{C}' = \{\langle b_i, \mathcal{B}'_i, d_i \rangle \mid \langle b_i, \mathcal{B}_i, d_i \rangle \in \mathcal{C}\}$ and $\mathcal{B}'_i = \{\langle L'_{ij}, U'_{ij} \rangle \mid \langle L_{ij}, U_{ij} \rangle \in \mathcal{B}_i$ and $L_{ij} \leqslant L'_{ij} \leqslant U'_{ij} \leqslant U_{ij}\}$.*

$C' = \{\langle b_i, \mathcal{B}'_i, d_i \rangle \mid \langle b_i, \mathcal{B}_i, d_i \rangle \in \mathcal{C}\}$ and $\mathcal{B}'_i = \{\langle L'_{ij}, U'_{ij} \rangle \mid \langle L_{ij}, U_{ij} \rangle \in \mathcal{B}_i$ and $L_{ij} \leqslant L'_{ij} \leqslant U'_{ij} \leqslant U_{ij}\}$.
Intuitively, given a DTNU $\mathcal{D}$ which is not weakly (resp. strongly) controllable, we want to find a DTNU $\mathcal{D}'$ which is weakly (resp. strongly) controllable. $\mathcal{D}'$ must be identical to $\mathcal{D}$ except for the bounds of the contingent constraints, which can be restricted (but not enlarged).

For example, consider a DTNU $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$ identical to the DTNU depicted in Figure 1b. This DTNU is not controllable, because the DTN projection $\mathcal{D}_\omega$ where $\omega(c_1) = 12$ and $\omega(c_2) = 4$ (c_1 being the contingent AC and c_2 the contingent DB) does not admit a consistent schedule. A solution to the weak-repair problem for $\mathcal{D}$ is a DTNU $\mathcal{D}' = \langle \mathcal{V}, \mathcal{E}, \mathcal{C}' \rangle$ such that c_1 is replaced by $c'_1 = (A, \{\langle 5, 8 \rangle\}, C)$. $\mathcal{D}'$ is one possible solution to the weak-repair problem.

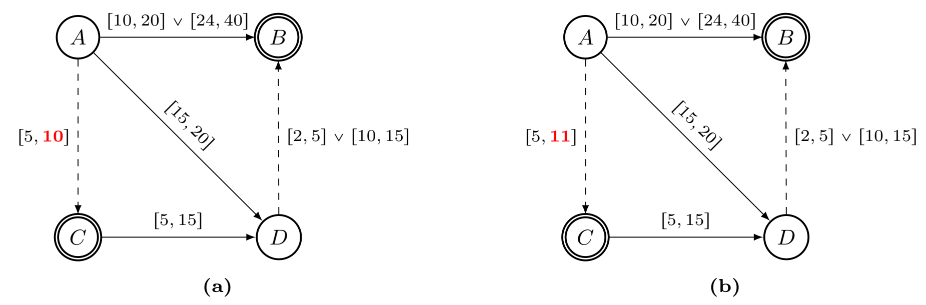


Fig. 2: Graph representations of the DTNU (a) an optimal solution to the strong-repair problem of $\mathcal{D}$, and DTNU (b) for the weak-repair problem

We are interested in repair solutions that minimize the reduction in the size of the contingent constraints' bounds. This intuitively corresponds to minimizing the flexibility for scheduling uncontrollable time points that are removed by the repair.

Definition 11. *Let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$, be a non τ-controllable DTNU and let $\mathcal{R}_{\mathcal{D}}$ be the set of all the solutions to the τ-repair problem for $\mathcal{D}$. An **optimal τ-repair** for D is defined as:*

$$\operatorname*{argmin}_{\mathcal{D}' \in \mathcal{R}_{\mathcal{D}}} \left(\sum_{\langle b_i, \mathcal{B}_i, d_i \rangle \in \mathcal{C}} \sum_{\langle L_{ij}, U_{ij} \rangle \in \mathcal{B}_i} ((L'_{ij} - L_{ij}) + (U_{ij} - U'_{ij})) \right)$$

We show in Figure 2, the optimal repair of $\mathcal{D}$ for strong and weak controllability.

5 DTNU Repair: SMT Encoding

The base encoding presented in Section 2 encodes a DTNU into an SMT model to check weak and strong controllability: in fact, all the encodings presented in [9, 7] are focused only on the problem of checking if a given DTNU is τ-controllable. To tackle the repair problem, we need to synthesize the new uncontrollable bounds. Hence, we need to extend the encoding to have the uncontrollable bounds as free variables so that the solver can assign concrete values to the uncontrollable bounds, as detailed below.

Compared to the basic encoding for controllability checking, we add two types of variables. We denote the original bounds of the i-th contingent constraints with L_{ij} and U_{ij} as in Definition 1. Then, for each such pair of contingent bounds, we introduce two free variables l_{ij} and u_{ij}; these represent the repaired bounds of the i-th contingent and will be constrained such that $L_{ij} \leqslant l_{ij} \leqslant u_{ij} \leqslant U_{ij}$. To simplify the notation, we indicate with $\vec{l}, \vec{u}$ the set of free variables for the lower and upper bounds, respectively. We redefine the formula for the contingent constraints in Definition 9 (indicated as $\Gamma'(\vec{y}, \vec{l}, \vec{u})$) which now depend not only on $\vec{y}$ but also on $\vec{l}$ and $\vec{\imath}$; instead, Ψ remains the same. With these new formulae, we will define the encodings for tackling the strong and weak repair problems for DTNU. In the following definition, we formalize these basic components of the encoding.

Definition 12. *Given a DTNU $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$, we define the following sets of SMT variables:*

- *$\vec{y}$ and $\vec{x}$ as in Definition 9.*
- *the* ***uncontrollable SMT lower bounds*** *$\vec{l}$ is a set of real SMT variables, one for each contingent constraint bound $L_{ij} \in \mathcal{B}_i$. We denote l_{ij}, the variable associated with L_{ij}.*
- *the* ***uncontrollable SMT upper bounds*** *$\vec{u}$ is a set of real SMT variables, one for each contingent constraint bound $U_{ij} \in \mathcal{B}_i$. We denote u_{ij}, the variable associated with U_{ij}.*

We also define the following SMT formulae:

- *$\Gamma'(\vec{y}, \vec{l}, \vec{u})$ representing the* ***SMT contingent constraints****:*

$$\Gamma'(\vec{y}, \vec{l}, \vec{u}) \doteq \bigwedge_{c_i \doteq \langle b_i, \mathcal{B}_i, d_i \rangle \in \mathcal{C}} \left(\bigvee_{\langle L_{ij}, U_{ij} \rangle \in \mathcal{B}_i} (y_i \geqslant l_{ij} \wedge y_i \leqslant u_{ij}) \right) \tag{5}$$

- *$\Psi(\vec{x}, \vec{y})$ as in Definition 9.*
- *$\Xi(\vec{l}, \vec{u})$ representing the* ***valid repair SMT constraints****:*

$$\Xi(\vec{l}, \vec{u}) \doteq \bigwedge_{\langle b_i, \mathcal{B}_i, d_i \rangle \in \mathcal{C}} \bigwedge_{\langle L_{ij}, U_{ij} \rangle \in \mathcal{B}_i} L_{ij} \leqslant l_{ij} \leqslant u_{ij} \leqslant U_{ij} \tag{6}$$

Considering the DTNU example in Figure 1b, we report the new variables and formulae below.

$$\vec{l} = \{l_{C1}, l_{B1}, l_{B2}\}$$
$$\vec{u} = \{u_{C1}, u_{B1}, l_{B2}\}$$

$$\Gamma'(\vec{y}, \vec{l}, \vec{u}) = (y_C \geqslant l_{C1}) \wedge (y_C \leqslant u_{C1}) \wedge$$
$$(((y_B \geqslant l_{B1}) \wedge (y_B \leqslant u_{B1})) \vee ((y_B \geqslant l_{B2}) \wedge (y_B \leqslant u_{B2})))$$

$$\Xi(\vec{l}, \vec{u}) = (5 \leqslant l_{C1} \leqslant u_{C1} \leqslant 10) \wedge (2 \leqslant l_{B1} \leqslant u_{B1} \leqslant 5) \wedge (10 \leqslant l_{B2} \leqslant u_{B2} \leqslant 15)$$

The new variable l_{C1} indicates the repaired lower bound of the contingent constraint $\langle A, \{\langle 5, 12 \rangle\}, C \rangle$; for the other contingent constraint in the example, we need two variables l_{B1} and l_{B2} because the constraint is disjunctive. Similarly for the upper bounds.

With these basic formulae, we can now define two encodings (called "quantified encodings"), one for the weak repair problem (Θ_{weak}) and one for the strong one (Θ_{strong}).

$$\Theta_{weak}(\vec{l},\vec{u}) \doteq \big(\forall \vec{y}.\exists \vec{x}.\Gamma'(\vec{y},\vec{l},\vec{u}) \rightarrow \Psi(\vec{x},\vec{y})\big) \wedge \Xi(\vec{l},\vec{u}) \tag{7}$$

$$\Theta_{strong}(\vec{x},\vec{l},\vec{u}) \doteq \big(\forall \vec{y}.\Gamma'(\vec{y},\vec{l},\vec{u}) \rightarrow \Psi(\vec{x},\vec{y})\big) \wedge \Xi(\vec{l},\vec{u}) \tag{8}$$

Note that for strong controllability, we do not need to quantify the controllable time points because by solving the SMT (or OMT) problem, we implicitly perform an existential quantification. Both encodings will have the property that all the models represent valid repairs for the input DTNU network, encoded in the values of the $\vec{l}$ and $\vec{u}$ variables. Suppose μ is a model for Θ_τ, then the DTNU with the new contingent constraints $\mathcal{C}'$ defined below is a valid τ-repair.

$$\mathcal{C}' \doteq \{\langle b_i, \{\langle \mu(l_{i,1}), \mu(u_{i,1})\rangle, \langle \mu(l_{i,2}), \mu(u_{i,2})\rangle, \ldots\}, d_i\rangle \mid \langle b_i, \mathcal{B}_i, d_i\rangle \in \mathcal{C}\}$$

Intuitively, we simply consider the value of l_{ij} in the model μ as the repaired value for L_{ij} and equivalently for the upper bounds.

Proposition 1. *Let $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C}\rangle$ be a DTNU such that $\mathcal{D}$ is not τ-controllable. Any model of Θ_τ yields a solution to the τ-repair problem and if the formula is unsatisfiable, the τ-repair problem admits no solution.*

It is easy to see why Proposition 1 holds: both the encodings are direct derivations from the ones for checking strong and weak controllability in [9] and [7], where we changed the constant bounds into SMT variables. Therefore, Proposition 1 follows from the correctness of these basic encodings.

Thanks to the guarantees of Proposition 1, we can use Optimization Modulo Theory to solve the optimal τ-repair problem by simply imposing an optimization objective over the τ-repair encodings.

$$\begin{aligned} &\text{minimize} \sum_{\langle b_i,\mathcal{B}_i,d_i\rangle \in \mathcal{C}} \sum_{\langle L_{ij},U_{ij}\rangle \in \mathcal{B}_i} ((l_{ij} - L_{ij}) + (U_{ij} - u_{ij})) \\ &\text{s.t. } \Theta_\tau \end{aligned} \tag{9}$$

This problem formulation can be solved by any OMT solver capable of dealing with LRA formulations such as Z3 [10]. Alternatively, it is possible to apply quantifier-elimination techniques to construct a quantifier-free formula equivalent to Ω_τ to be used by any OMT solver for QF_LRA.

6 STNU Repair: SMT Encoding

In this Section, we tackle the problem of temporal network repair in the special case of STNU. In fact, the general encoding for DTNU requires the use of quantifiers that are decidable in LRA but bring a very high computational cost. To optimize the encoding for STNU, we removed the quantification of the variables in $\vec{y}$ exploiting the convexity of the problem, as proved in [25], considering all combinations of the lower and upper bounds of the contingents is enough to check the controllability of an STNU. This allows us to fix the duration of contingent constraints to either their lower bound (in $\vec{l}$) or upper bound (in $\vec{u}$) and disregard the values within the interval because the problem convexity guarantees that the problem is controllable if it is controllable "on the bounds." In the following, we formalize the basic components of the encoding for STNU.

Definition 13. *Given an STNU $\mathcal{D} = \langle \mathcal{V}, \mathcal{E}, \mathcal{C} \rangle$, we define the following sets of SMT variables:*

- *$\vec{x}$ as in Definition 9.*
- *$\vec{l}$ and $\vec{u}$ as in Definition 12*
- *the **set of symbolic boundary projections** $\mathcal{P}(\vec{l}, \vec{u})$ is a set of vectors of SMT variables, one for each possible combination of lower and upper bound for each contingent constraint (hence a total of $2^{\mathcal{C}}$ vectors):*

$$\mathcal{P}(\vec{l}, \vec{u}) = \{\vec{v} \mid v_i \in \{l_{i1}, u_{i1}\}, c_i \in \mathcal{C}\}$$

- *a set of "fresh" real SMT variables[3], one for each element $\vec{v}$ of $\mathcal{P}(\vec{l}, \vec{u})$, indicated as $\vec{x}^{\vec{v}}$. We indicate the set of all these free variables as $\vec{x}^{\mathcal{P}}$.*

By exploiting the convexity of the STNU fragment, we can re-formulate the encodings of the strong and weak repair as follows.

$$\Theta_{weak}(\vec{x}^{\mathcal{P}}, \vec{l}, \vec{u}) = \left(\bigwedge_{\vec{v} \in \mathcal{P}(\vec{l}, \vec{u})} \Psi(\vec{x}^{\vec{v}}, \vec{v}) \right) \wedge \Xi(\vec{l}, \vec{u}) \tag{10}$$

$$\Theta_{strong}(\vec{x}, \vec{l}, \vec{u}) = \left(\bigwedge_{\vec{v} \in \mathcal{P}(\vec{l}, \vec{u})} \Psi(\vec{x}, \vec{v}) \right) \wedge \Xi(\vec{l}, \vec{u}) \tag{11}$$

We call this the "on-bounds" encoding. The key intuition here is that for the STNU fragment, we can ensure the controllability of the network by checking that a strategy for scheduling controllable time points exists for every possible combination of the bounds. The difference between strong and weak controllability here becomes evident: in Θ_{strong} we have one variable for each controllable time point ($\vec{x}$), and we need a single assignment that works for all the combinations of lower and upper bounds; in weak controllability we have one vector of controllable time points variables ($\vec{x}^{\vec{v}}$) for every combination $\vec{v}$ of lower and upper bounds, and therefore the solver can assign different values to the controllable for every bound combination.

[3] In SMT jargon, a fresh variable is a new variable with a name that is unused in the rest of the encoding and with no additional constraints.

We remark that this encoding only works in the STNU fragment but retains all the properties of Proposition 1 and can be used for solving the optimal repair problem analogously to the DTNU case.

Finally, we remark that another strength of our proposal is that both the quantified and the on-bounds encodings can compute a schedule for the controllable time points in the strong controllability case.

7 Experiments

In this Section, we empirically evaluate the effectiveness of the proposed approaches. We implemented all the encodings in Python using the pySMT framework [12]. For our experiments, we use the Z3 solver as the backend.

We experimented on a large set of DTNU and STNU benchmarks. For the DTNU case, we consider the benchmark set of random networks used by Osanlou et al. in [18], which is composed of 1500 DTNU, 500 of them of the size between 10 to 20, 500 between 20 to 25, and 500 between 25 to 30 time points.

For the STNU case, we implemented a custom random generator: we create an STNU in the form of a complete directed acyclic graph (DAG); then, we randomly and safely remove some of the edges according to the following parameters:

- the number of time points n;
- the number of uncontrollable time points set by a range $m = [min_m, max_m]$, i.e., it creates an STNU with $|\mathcal{V}_u| = k$ with $k \in [min_m, max_m]$;
- the contingency rate per edge, i.e., the probability for a constraint to become a contingent. We keep this rate low to scatter the contingent in the graph (less than 10 %);
- the rate of removal q, i.e., the probability for a constraint to be removed representing the sparsity of the graph.

We generated STNUs of different sizes: $n \in \{5, 10, 20, 50, 100\}$ and with different degrees of uncontrollability depending on the size: $m = \{[1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [8, 10]\}$. The combinations are as follows: n=5 with {[1,2], [2,3]}, n=10 with {[2,3], [3,4]}, n=20 with {[3,4], [4,5]}, n=50 with {[5,6], [8,10]}, and n=100 with {[8,10]}. Finally, for each relevant combination of n and m, we change the density of the graph with $q = \{0\%, 30\%, 50\%, 75\%, 90\%\}$ to get a benchmark set of 1100 non-controllable STNUs.

We performed a total of 7400 tests: 2*1500 for weak and strong repairs on DTNUs and 4*1100 on STNUs for both the quantified and on-bounds encodings and both controllability levels. We ran all the experiments on an Xeon E5-2620 2.10GHz with 3600s/10GB time/memory limits.

Figure 3 shows the results of the encodings for DTNUs (in logarithmic scale) with the number of instances solved on the x-axis and the time on the y-axis. The plot shows that the strong-repair problem is much easier to solve for our encodings than the weak-repair: we could solve the strong-repair for all the instances, while the weak one approximately solves the easiest 500 instances.

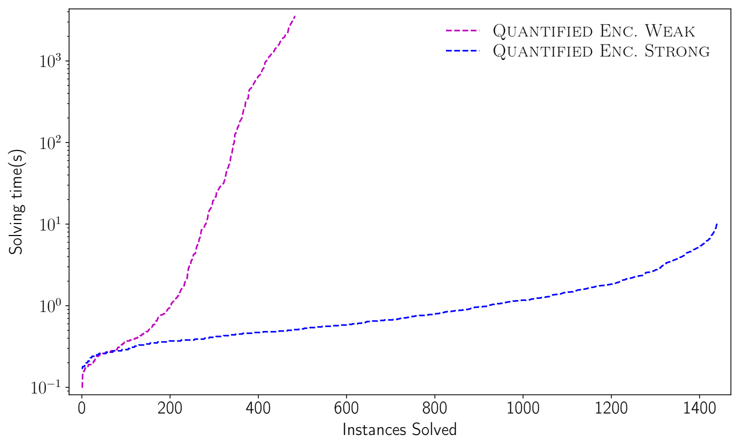


Fig. 3: DTNU cactus plot of weak and strong repair: for each encoding, we plot the solving time for the instances sorted from the easiest to the hardest.

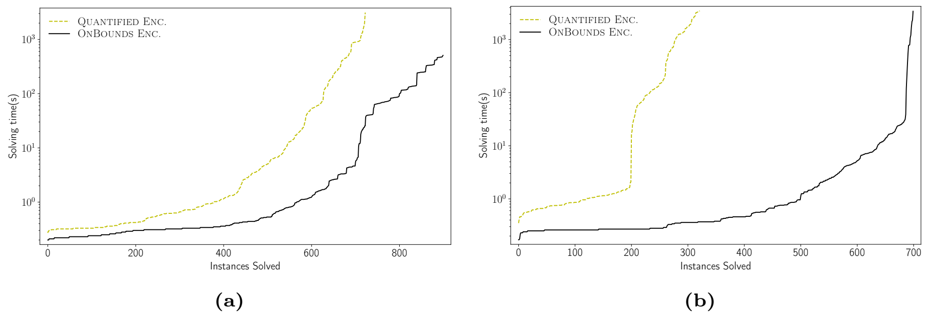
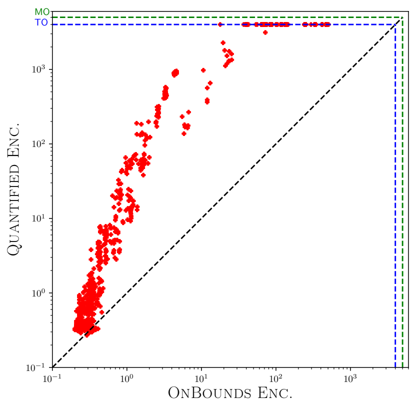


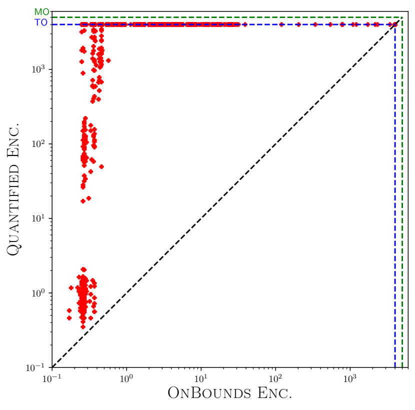
Fig. 4: STNU cactus plots for the strong (a) and weak (b) repair problems.

This is not surprising, as in Θ_{strong}, we only have one existential quantifier, while in Θ_{weak}, we have a quantifier alternation.

The cactus plot for STNU instances is shown in Figure 4. We compare the quantified and on-bounds encodings for strong (left) and weak (right) repair. In general, the on-bounds encoding for STNU can solve many more instances than the quantified one. In Figure 5, we also report scatter plots for strong and weak repair that highlight the dominance of the on-bounds encoding for this class of problems. Moreover, as shown in the scatter plots, we never hit the memory limit with our encodings, the limiting dimension is time, while memory consumption is not an issue (by manual inspection, we report that we never exceed 2GB of memory usage).



(a)



(b)

Fig. 5: STNU scatter plots for the strong (a) and weak (b) repair problems.

8 Conclusions and future work

In this paper, we tackled the repair problem in temporal networks under uncertainty. The problem is defined as finding a variant of an uncontrollable network that is obtained by reducing the bounds of the contingent constraints and is controllable. We presented two SMT-based algorithms, tackling the specific problems of Weak and Strong controllability. The first one is a general encoding for DTNUs that works for any controllability that can be encoded into SMT. The second one efficiently tackles the repair problem for the weak controllability of STNU by exploiting the problem convexity. We also define and solve using OMT the optimal repair problem, that is finding the repair that removes the least amount of flexibility from the original temporal network. The experimental evaluation shows that the general case is heavily dependent on the number of uncontrollables and that the STNU specialized algorithm leads to substantial increases in efficiency.

For future work, we plan to explore the possibility of tackling the STNU repair problem using constraint propagation techniques. Using propagation algorithms, the checking problem for SC and DC on STNU has proven to be polynomial. Such algorithms can infer a negative cycle if the STNU is not controllable. Thus, fixing all the negative cycles incrementally might be more efficient. Moreover, we plan to apply these ideas to a multi-agent setting where an agent's contingent constraint might be controllable decisions for another, and the repair problem can serve as a means for negotiation among agents.

Acknowledgements

Andrea Micheli and Alessandro Cimatti have been partly supported by the PNRR project FAIR - Future AI Research (PE00000013) and by the Interconnected Nord-Est Innovation Ecosystem (iNEST) (ECS00000043), under the NRRP MUR program funded by the NextGenerationEU. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

[1] Shyan Akmal et al. "Quantifying controllability in temporal networks with uncertainty". In: *Artificial Intelligence* 289 (2020), p. 103384. ISSN: 0004-3702.

[2] Shyan Akmal et al. "Quantifying degrees of controllability in temporal networks with uncertainty". In: *Proceedings of the International Conference on Automated Planning and Scheduling*. Vol. 29. 2019, pp. 22–30.

[3] Clark Barrett and Cesare Tinelli. *Satisfiability modulo theories*. Springer, 2018.

[4] Claudio Bettini, X Sean Wang, and Sushil Jajodia. "Solving multi-granularity temporal constraint networks". In: *Artificial Intelligence* 140.1-2 (2002), pp. 107–152.

[5] Nikolaj Bjørner, Anh-Dung Phan, and Lars Fleckenstein. "νz-an optimizing SMT solver". In: *Tools and Algorithms for the Construction and Analysis of Systems: 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings 21*. Springer. 2015, pp. 194–199.

[6] Guillaume Casanova et al. "Solving Dynamic Controllability Problem of Multi-Agent Plans with Uncertainty Using Mixed Integer Linear Programming". In: *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*. Ed. by Gal A. Kaminka et al. Vol. 285. Frontiers in Artificial Intelligence and Applications. IOS Press, 2016, pp. 930–938.

[7] Alessandro Cimatti, Andrea Micheli, and Marco Roveri. "An SMT-based approach to weak controllability for disjunctive temporal problems with uncertainty". In: *Artificial Intelligence* 224 (2015), pp. 1–27.

[8] Alessandro Cimatti, Andrea Micheli, and Marco Roveri. "Dynamic controllability of disjunctive temporal networks: Validation and synthesis of executable strategies". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1. 2016.

[9] Alessandro Cimatti, Andrea Micheli, and Marco Roveri. "Solving strong controllability of temporal problems with uncertainty using SMT". In: *Constraints* 20 (2015), pp. 1–29.

[10] Leonardo De Moura and Nikolaj Bjørner. "Z3: An efficient SMT solver". In: *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2008, pp. 337–340.

[11] Rina Dechter, Itay Meiri, and Judea Pearl. "Temporal constraint networks". In: *Artificial intelligence* 49.1-3 (1991), pp. 61–95.

[12] Marco Gario and Andrea Micheli. "pySMT: a Solver-Agnostic Library for Fast Prototyping of SMT-Based Algorithms". In: *SMT Workshop*. 2015.

[13] Luke Hunsberger and Roberto Posenato. "Speeding Up the RUL Dynamic-Controllability-Checking Algorithm for Simple Temporal Networks with Uncertainty". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 9. 2022, pp. 9776–9785.

[14] Josef Lubas, Marco Franceschetti, and Johann Eder. "Resolving conflicts in process models with temporal constraints". In: *Proceedings of the ER Forum and PhD Symposium*. 2022.

[15] Andrea Micheli. "Disjunctive temporal networks with uncertainty via SMT: Recent results and directions". In: *Intelligenza Artificiale* 11.2 (2017), pp. 155–178.

[16] Paul H. Morris and Nicola Muscettola. "Temporal Dynamic Controllability Revisited". In: *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*. Ed. by Manuela M. Veloso and Subbarao Kambhampati. AAAI Press / The MIT Press, 2005, pp. 1193–1198. URL: `http://www.aaai.org/Library/AAAI/2005/aaai05-189.php`.

[17] Paul H. Morris, Nicola Muscettola, and Thierry Vidal. "Dynamic Control Of Plans With Temporal Uncertainty". In: *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001, Seattle, Washington, USA, August 4-10, 2001*. Ed. by Bernhard Nebel. Morgan Kaufmann, 2001, pp. 494–502.

[18] Kevin Osanlou et al. "Solving Disjunctive Temporal Networks with Uncertainty under Restricted Time-Based Controllability Using Tree Search and Graph Neural Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36 (2022), pp. 9877–9885.

[19] Roberto Posenato and Carlo Combi. "Adding flexibility to uncertainty: Flexible simple temporal networks with uncertainty (FTNU)". In: *Information Sciences* 584 (2022), pp. 784–807.

[20] Roberto Sebastiani and Silvia Tomasi. "Optimization Modulo Theories with Linear Rational Costs". In: *ACM Trans. Comput. Log.* 16.2 (2015), 12:1–12:43.

[21] Julie A Shah and Brian Charles Williams. "Fast Dynamic Scheduling of Disjunctive Temporal Constraint Networks through Incremental Compilation." In: *ICAPS*. 2008, pp. 322–329.

[22] Ilia Stepin et al. "A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence". In: *IEEE Access* 9 (2021), pp. 11974–12001.

[23] Kostas Stergiou and Manolis Koubarakis. "Backtracking algorithms for disjunctions of temporal constraints". In: *Artificial Intelligence* 120.1 (2000), pp. 81–117.

[24] Kristen Brent Venable and Neil Yorke-Smith. "Disjunctive Temporal Planning with Uncertainty". In: *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005.* Professional Book Center, 2005, pp. 1721–1722.

[25] Thierry Vidal and Hélène Fargier. "Handling contingency in temporal constraint networks: from consistency to controllabilities". In: *Journal of Experimental & Theoretical Artificial Intelligence* 11.1 (1999), pp. 23–45.