Study of the Computational Complexity of the Repair Problem on Simple Temporal Networks with Uncertainty

Frédéric Maris[1,2], Aïdin Sumic[3], Thierry Vidal[4], Bruno Zanuttini[1]

[1] Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ
GREYC UMR6072, F-14000 Caen, France
[2] Université de Toulouse, IRIT
[3] ONERA de Toulouse, DTIS
[4] Université de Technologie Tarbes Occitanie Pyrénées, PICS

frederic.maris@irit.fr, aidin.sumic@onera.fr, thierry.vidal@uttop.fr, bruno.zanuttini@unicaen.fr

Résumé

Les réseaux temporels simples avec incertitude (STNU) sont un modèle basé sur les contraintes conçu pour vérifier la contrôlabilité temporelle d'un plan sous incertitude (lorsque certaines durées contingentes ne peuvent être déterminées par l'agent de planification). Quoi qu'il en soit, dans les cas de planification multi-agents ou dépendante des ressources, il peut être possible de négocier ou de reconsidérer la flexibilité exogène et d'ajuster les limites des durées incontrôlables. Plusieurs travaux proposent une solution optimale à ce problème de réparation selon les trois niveaux de contrôlabilité (faible, dynamique, forte). Cependant, aucun d'entre eux ne caractérise la complexité théorique de ce problème. Cet article propose une évaluation approfondie de la complexité du problème de réparation pour les STNU.

Mots-clés

Complexité, satisfaction de contraintes, incertitude temporelle, réseaux temporels.

Abstract

Simple Temporal Networks with Uncertainty (STNU) are a constraint-based model designed to check the temporal 'controllability' of a plan under uncertainty (when some contingent durations cannot be decided by the planning agent). Anyway, in multi-agent or resource-dependent planning cases, it may be possible to negotiate or reconsider exogenous flexibility and adjust the bounds of the uncontrollable durations. Several works exist that provide the optimal solution to this repair problem depending on the three controllability levels (Weak, Dynamic, Strong). However, none of them characterize the theoretical complexity of this problem. This paper will provide a thorough complexity evaluation of the repair problem for STNU.

Keywords

Complexity, constraint satisfaction, temporal uncertainty, temporal networks.

1 Introduction

In many domains one needs to reason on activities that may or must not overlap in time, last over some duration, and synchronize with timestamped expected events. That is particularly true in planning and scheduling, where existing systems often use some explicit constraint-based representation [6, 18, 17, 4].

An efficient model for managing temporal durations is the Simple Temporal Network (STN) [8]: nodes are timepoints, and edges are constraints expressing convex intervals of possible durations between them. STNs are widely used to check a plan's temporal *consistency*. Consistency checking is made through polynomial-time propagation algorithms (e.g., the Floyd-Warshall reduction) and provides a complete *minimal* network in which all inconsistent values are removed [8]. This minimal network can be passed on to the plan execution manager, which can take any value on the domain of the first activity to schedule, repropagate, and so on iteratively.

A well-known extension of STNs that handles uncertainties, called STNU (Simple Temporal Network with Uncertainty), has been proposed by [19]. An STNU contains uncertain (*contingent*) durations between time-points, which means the effective duration is not controlled by the agent executing the plan. This is useful for addressing realistic dynamic and stochastic domains.

Temporal consistency is then redefined as *controllability*: an STNU is controllable if a strategy exists for executing the plan, whatever the values taken by the contingent durations. Three levels of controllability exist that depend on when the contingents' duration will be known: just before execution (*Weak*), during execution (*Dynamic*), or never (*Strong*).

Checking is one thing, but what to do when a plan is temporally uncontrollable has received less attention. There are many ways to respond to that at different levels. The highest is replanning, i.e., reconsidering the course of actions leading to the goal. Then, one may only reason at the scheduling level: switching the order of mutually exclusive actions, swapping a resource for another faster one, etc; such changes are lighter and could be sufficient to change some durations for the better. Last, the lightest thing to do is only to alter the temporal durations themselves, if possible. But the durations that the executing agent controls have already been shrunk as much as possible by propagation algorithms (as in any constraint-based model), while altering a contingent duration is impossible and even forbidden as it is controlled by some other entity, often *Nature* itself.

In [15], the authors argue that contingents may be reduced in some applications in which their durations depend on the plan quality measured through some *utility* of chosen actions. Hence, they proposed some algorithms that reduce contingents to recover controllability (of STNUs) even if it results in solutions of lesser quality.

For instance, the uncertain duration of a data downloading for an observation satellite may depend on the (exogenous) imprecise size of such data: forcing a lower upper bound is possible, meaning sending incomplete or less detailed data in extreme cases, resulting in a lower quality of the plan. Another example is in manufacturing cells in which machines are set up (speed, tools) to provide the best quality, and production plans are generated in such a framework; but it is still possible in case of temporally uncontrollable plans to change those settings, e.g., by authorizing higher speed, at the price of a lower quality: in section 4 we will provide a more precise example under such assumptions.

In other words, *Nature* there appears to be some external entity, not involved into the planning and executing processes, but over which some preprocessing control is still possible.

More interesting is the case of multi-agent planning, as replanning is a last-resort solution there, as it will impinge on the coordinated execution of the whole group; the most local and least committed repair will be sought. Moreover, uncertainty often comes from the decisions of other agents: an activity duration being decided by some other agent, but shared just before execution, or during execution, or not communicated at all. In that regard, the authors in [16] proposed a new multi-agent model for STNUs that define such a particular constraint called *contract*, and proposed some algorithms to repair any non-controllable STNU by reducing these *contracts*.

These scenarios show the relevance of the STNU Repair problem (but also in higher classes of Temporal Networks under Uncertainty). However, there exists no formal proof of its computational complexity: this paper aims to fill the gap.

In the remainder we first expose the necessary background on STNUs and some related work in Sections 2 and 3. Then, in Sections 4 and 5, we characterize the STNU Repair problems in a new way and based on that we present our study of their complexities. Next, we will discuss the case of repairing for multiple agents in Section 6. Finally, we will conclude our study with some prospects.

2 Background

2.1 Simple Temporal Network

A Simple Temporal Network (STN) [8] is a tuple $\langle \mathcal{V}, E \rangle$, where $\mathcal{V}$ is a set of real-value variables called timepoints $\{v_0, \ldots, v_n\}$, and E is a set of temporal constraints between these timepoints called *requirements* [8]. Specifically, each requirement constraint $e_k \in E$ is of the form $v_j - v_i \in [l, u]$, with $v_i, v_j \in \mathcal{V}$, $l \in \mathbb{R} \cup \{-\infty\}$, and $u \in \mathbb{R} \cup \{+\infty\}$. l and u represent the minimal and maximal possible temporal distance between v_i and v_j. A requirement constraint can be represented also as an edge $v_i \xrightarrow{[l,u]} v_j$.
A reference timepoint v_0 is generally included in $\mathcal{V}$ as a fixed temporal anchor for all other timepoints. v_0 is assumed to be the first executed timepoint of the network, i.e., there is an implicit constraint $v_0 \leqslant v_i$ for each $v_i \in \mathcal{V}$.
An STN is *consistent* (i.e., satisfiable) if an assignment (schedule) to timepoints exists that satisfies all the constraints. We say that a controller executes an STN when it schedules its timepoints. Consistency checking is done in polynomial time and achieved by transforming an STN into a *distance graph*. A distance graph is a directed graph where each requirement constraint ($v_i \xrightarrow{[l,u]} v_j$) is split into two edges: $v_i \xrightarrow{u} v_j$ and $v_j \xrightarrow{-l} v_i$ allowing positive and negative weights. An STN is inconsistent if its distance graph contains a negative cycle, which can be detected using shortest-path algorithms.

2.2 STNs with Uncertainty

An STN with Uncertainty (STNU) is an extension of STNs in which one distinguishes a subset of constraints whose values (duration) are decided by external entities (i.e., *Nature*) and the controller can only observe [19].

Definition 1 (STNU)**.** *An STNU is a tuple $\langle \mathcal{V}, E, C \rangle$ with:*

- *$\mathcal{V} = \mathcal{V}_c \cup \mathcal{V}_u$ is a set of timepoints: $\mathcal{V}_c$, the set of controllable timepoints, and $\mathcal{V}_u$, the set of contingent (uncontrollable) ones.*
- *E is a set of requirement constraints as in an STN.*
- *C is a set of contingent links. A contingent link c_k is of the form $v_j - v_i \in [l, u]$, where $v_i \in \mathcal{V}_c$, $v_j \in \mathcal{V}_u$, and $0 \leqslant l \leqslant u$. The value $\omega_k = v_j - v_i$ is called* duration*, and an external entity decides it before or during execution. Once v_i is executed, the value of v_j is $v_i + \omega_k$. Any c_k is also depicted as $v_i \overset{[l,u]}{\dashrightarrow} v_j$.*

Definition 2 (Schedule)**.** *A **schedule** for an STNU $\mathcal{X}$ is a mapping $\delta\colon \mathcal{V} \to \mathbb{R}$ from timepoints to real values.*

Definition 3 (Situation, Projection and Solution)**.** *Given an STNU $\mathcal{X} = \langle \mathcal{V}, E, C \rangle$, the **situations** of $\mathcal{X}$ is a set of tuples Ω defined as the Cartesian product of contingent domains: $\Omega = \times_{c_k \in C}[l, u]$. Each situation $\omega = \{\omega_1, \ldots, \omega_n\} \in \Omega$ represents one possible complete set of values for the duration of the contingent links of $\mathcal{X}$. A **projection** $\mathcal{X}^\omega = (\mathcal{V}, E \cup C^\omega)$ of $\mathcal{X}$ is an STN where $C^\omega = \{[\omega_k, \omega_k] \mid c_k \in$

*$C\}$. A **solution** of $\mathcal{X}^{\omega}$ is a schedule δ_{ω} satisfying all the constraints.*

Intuitively, a projection replaces each contingent link with a rigid requirement constraint, i.e., a constraint reduced to a one-value range associated with each contingent link in ω.
In STNUs, consistency needs to be redefined: a network is now said to be *controllable* if there exists a schedule that satisfies all requirement constraints in any possible projection.

Definition 4 (Decision and Observation)**.** *$\forall v_i \in \mathcal{V}_c$, $dec(v_i)$ is the instant at which $\delta(v_i)$ is **decided** by the controller.*
*$\forall \omega_k \in C$, $obs(\omega_k)$ is the instant at which ω_k is **observed** by the controller.*

The controllability properties were defined in [19], and their semantics refined in [16]: *Strong Controllability (SC)* assumes one common schedule δ satisfies all constraints for all situations, which is relevant when external events durations cannot be observed (conformant planning). On the contrary, *Weak Controllability* (WC) assumes there is one consistent schedule for each situation, which is relevant when all contingent durations will be known just before execution (oracle). In-between, the *Dynamic Controllability (DC)* assumes the schedule values assignment depends on past observations only, regardless of the contingent durations still to be observed.

Definition 5 (Weak Controllability (WC))**.** *An STNU $\mathcal{X}$ is **weakly controllable** iff $\forall \omega \in \Omega, \exists \delta$ s.t. δ_{ω} is a solution of $\mathcal{X}_{\omega}$.*
Execution semantics: $\forall \omega_k \in \omega$, $obs(\omega_k) = v_0$, and the decision policy is free: $\forall v_i \in \mathcal{V}_c$, $dec(v_i) \leqslant v_i$.

Definition 6. ***(Strong Controllability (SC) with Execution)** An STNU $\mathcal{X}$ is **strongly controllable** iff $\exists\, \delta$ such that $\forall \omega \in \Omega, \delta$ is a solution of $\mathcal{X}_{\omega}$.*
Execution semantics: $\forall v_i \in \mathcal{V}_c$, $dec(v_i) = v_0$, and the observations are free: possibly no observation ($\forall \omega_k \in \omega$, $obs(\omega_k) = \emptyset$) or observations during execution that will just update the bounds of the constraints in the network.

Definition 7. ***(Dynamic Controllability (DC) with Execution)** An STNU $\mathcal{X}$ is **Dynamically controllable** iff it is Weakly controllable and $\forall v_i \in \mathcal{V}_c, \forall \omega, \omega' \in \Omega$, $\omega^{\leq v_i} = \omega'^{\leq v_i} \implies \delta_{\omega}(v_i) = \delta'_{\omega}(v_i)$*
where $\omega^{\leq v} = \{\omega_k \in \omega$ s.t. $obs(\omega_k) \preceq dec(v)\}$ is the part of the situation ω in which contingent constraints values are observed before executing v.
Execution semantics: $\forall \omega_k \in \omega$, $obs(\omega_k)$=$end(c_k)$, and $\forall v_i \in V_c$, $dec(v_i) = v_i$

Dynamic and Strong Controllability have proven to be retractable and solvable in polynomial time [19, 10], while WC has been proven to be a co-NP complete problem [12]. In fact, DC and WC checking rely on algorithms that look for negative cycles in the *distance graph* of STNUs [10, 15].

3 Related Work

The notion of repair arises in [5]. They introduced a Multi-agent STNU (MaSTNU) model in which agents have their own plans, and hence own temporal networks, but all face common exogenous contingent constraints. They proposed a new algorithm for checking the Dynamic Controllability (DC) of MaSTNU using a Mixed Integer Linear Programming (MILP) approach. The authors claim that it should be possible to modify the encoding to reduce the bounds of the contingents so that all 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 Strong and Dynamic Controllability. Later, they propose an incomplete Linear Programming approach to repair a non-DC STNU by repairing the negative cycles [3]. However, it is incomplete because it assumes that negative cycles are independent, which is not always the case.
Recently, the repair problem for STNU was formally defined in [15]. The authors tackle the case of WC and SC through Satisfiability Modulo Theory (SMT) and propose an optimization function that finds the minimal reduction of the contingent bounds to repair the network.
Later on, they proposed a distributed extension of STNUs called Multi-Agent Interdependent STNUs (MISTNU) that introduced the notion of *contract*, a shared constraint being controllable for one agent but contingent for others [16]. Hence, a contingent, being now actually controlled by another agent of the system (that is not *Nature*), becomes negotiable, making the repair problem of STNU clearly more relevant than in MaSTNU. In addition, they extend the SMT encodings to repair non-controllable MISTNU.
However, such encodings are sensitive to the number of contingents/contracts, hence, poorly scalable. Therefore, they argue that using propagation-based algorithms to identify and repair sources of uncontrollability should be more efficient and scalable.
In that regard, the closest related works that focus on STNUs and diagnosis, i.e., pinpointing reasons for non-controllability, address DC [11] and WC [14] by providing new checking algorithms that are informed, i.e., that return the sources of uncontrollability in the form of negative cycles in the distance graph of STNU. Please note that the former approach [11] for DC returns only one negative cycle at a time. Hence, one must use an iterative process to repair all of them. Yet, no repair algorithm currently exists that is based on these informed algorithms.
In this paper, we are interested in evaluating the theoretical complexity of the repair problem. Therefore, the following section will generalize the previous studies by introducing different types of repair problems relevant enough for STNUs to determine their complexity.

4 Repair Problems

In this section, we formally define the repair problems. In the following definitions, τ stands for any controllability level in $\{\mathsf{S}, \mathsf{D}, \mathsf{W}\}$ (standing for "Strong", "Dynamic", and "Weak", respectively). We start with some new fundamental notions.

Definition 8 (Tightening and τ-repair)**.** *Let $\mathcal{X} = \langle \mathcal{V}, E, C \rangle$ be an STNU. A **tightening** of $\mathcal{X}$ is a set of ordered pairs $\rho = \langle\langle c_1, [l'_1, u'_1]\rangle, \ldots, \langle c_k, [l'_k, u'_k]\rangle\rangle$ such that $c_1, \ldots, c_k \in C$ are pairwise distinct contingent constraints of $\mathcal{X}$ and for $i = 1, \ldots, k$, if c_i is $v \xdashrightarrow{[l_i, u_i]} v'$, then $l_i \leqslant l'_i \leqslant u'_i \leqslant u_i$ holds. We write $\mathcal{X} \oplus \rho$ for the STNU $\langle \mathcal{V}, E, C' \rangle$, where C' is obtained from C by replacing each c_i by $v \xdashrightarrow{[l'_i, u'_i]} v'$.*
A τ-repair is a tightening such that $\mathcal{X} \oplus \rho$ is τ-Controllable.

Definition 9 (Repair Problem)**.** τ-REPAIR *is the decision problem:*

- Input: *An STNU $\mathcal{X}$*
- Question: *Is there a τ-repair ρ of $\mathcal{X}$?*

Definition 9 introduces a basic formulation of the repair problem. However, this minimal definition may not be sufficient in more realistic settings. Depending on the environment, such as the influence of *Nature* or other agents, or on specific optimization criteria and performance metrics relevant to the agent's objectives, a more refined definition may be required.
In order to define such refined definitions, we first introduce additional notation; let $\mathcal{X}$ be an STNU, $\rho = \langle\langle c_1, [l'_1, u'_1]\rangle, \ldots, \langle c_k, [l'_k, u'_k]\rangle\rangle$ a tightening of $\mathcal{X}$, and l_j, u_j the lower and upper bounds of c_j in $\mathcal{X}$; then we write

- $\mathrm{Supp}(\rho)$ for the set $\{j \in \{1, \ldots, k\} \mid l'_j \neq l_j \vee u'_j \neq u_j\}$, i.e., the set of constraints actually (strictly) repaired by ρ;
- $\mathrm{Cost}(\rho)$ for the quantity $\sum_{j=1,\ldots,k} l'_j - l_j + u_j - u'_j$, i.e., the sum of all interval reductions over all contingent constraints.

Definition 10 (Partial Repair)**.** PARTIAL-τ-REPAIR *is the following decision problem:*

- Input: *An STNU $\mathcal{X}$ and a subset R of its contingent constraints*
- Question: *Is there a τ-repair ρ of $\mathcal{X}$ such that $\mathrm{Supp}(\rho) \subseteq R$?*

The partial repair problem is relevant when only a subset of contingent durations can be reduced, such as in multi-agent settings where some durations are negotiable (e.g, belong to another agent) while others, controlled by *Nature*, must remain fixed.

Definition 11 (k-Budget Repair)**.** k-BUDGET-τ-REPAIR *is the following decision problem:*

- Input: *An STNU $\mathcal{X}$ and a rational number k*
- Question: *Is there a τ-repair ρ of $\mathcal{X}$ with $\mathrm{Cost}(\rho) \leqslant k$?*

The *k-budget* repair problem is particularly relevant in scenarios where optimizing a specific parameter is important, such as minimizing cost [7, 20], or maximizing flexibility or fairness among agents in multi-agent settings [16].

Definition 12 (k-constraint repair)**.** k-CONSTRAINT-τ-REPAIR *is the following decision problem:*

- Input: *An STNU $\mathcal{X}$ and an integer k*
- Question: *Is there a τ-repair ρ of $\mathcal{X}$ with $|\mathrm{Supp}(\rho)| \leqslant k$?*

The *k-constraint* repair problem is also relevant for optimization in multi-agent settings, where minimizing communication costs amounts to looking for the smallest set of contingents to negotiate.
To better grasp the STNU model and the different repair problems, we provide a realistic example, inspired by the application framework discussed in the Introduction, in Example 1.

Example 1. *Bob operates on a production line involving two machines, M_1 and M_2. Each day, a manager determines the production requirements following predefined regulations. The process begins with M_1 transforming raw materials into components. Bob then performs a quality check before passing the components to M_2, which completes the final product. The durations of tasks executed by M_1 and M_2 are uncertain, ranging from 5–12 minutes and 10–15 minutes, respectively, while Bob's task is controllable and takes between 5–15 minutes. The overall process must be completed within a time window of 25 to 30 minutes. To guarantee feasibility, the manager may adjust the machines' speed before execution, but at a cost.*

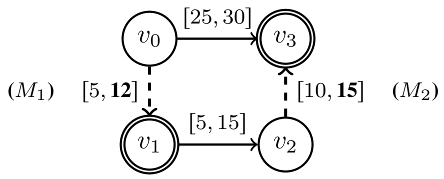
Figure 1 illustrates Bob's scheduling problem. The STNU is not weakly controllable, and a repair solution is shown with new bounds on M_1 and M_2. WC semantics is chosen here for its simplicity.

5 Complexity of Repair Problems

This section provides complexity results for the different types of repair problem (see Definitions 9, 10, 11 and 12) according to the semantics of the three controllability levels. The results of our study are summarized in Table 1.

Problem	Strong	Dynamic	Weak
Controllability	P [19]	P [10]	coNP-comp [12]
Repair	P	P	P
Partial repair	P	in NP	coNP-hard and in Σ_2^P
k-budget repair	P	in NP	coNP-hard and in Σ_2^P
k-constraint repair	NP-comp	NP-comp	coNP-hard and in Σ_2^P

Table 1: Complexity of controllability and repair problems ("comp" is short for "complete").



c_k	Repair	Partial R. (M_1)	2-Budget R.	1-Cons. R.
M_1	**[10, 10]**	**[7, 7]**	**[5, 11]**	[5, 12]
M_2	**[10, 10]**	[10, 15]	**[10, 14]**	**[10, 10]**

Figure 1: STNU of Example 1 where nodes represent timepoints (doubly circled ones are uncontrollable), edges are requirements and contingent constraints (the later relating to the tasks of machines M_1 and M_2). The STNU is not WC due to the highlighted projection $\omega = \{\omega_1 = 12, \omega_2 = 15\}$. The table highlights a solution to each type of repair problem.

5.1 Complexity of the Repair Problem

We study here the repair problem from definition 9, i.e., without an optimization criterion. We first introduce a straightforward lemma, which shows that when repairs are unconstrained, we can consider only repairs to singleton intervals without loss of generality, and additionally that we can restrict to rational numbers, of size polynomial in the size of the STNU. In fact, this is the same as considering a discretization of the intervals of the contingents with a polynomial number of possible values.

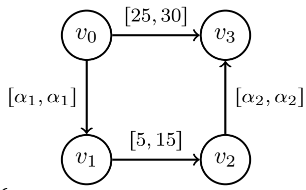
Lemma 1. *Let $\tau \in \{\mathsf{S}, \mathsf{D}, \mathsf{W}\}$ be a level of controllability. If an STNU $\mathcal{X}$ has a τ-repair $\langle\langle c_1, [l'_1, u'_1]\rangle, \ldots, \langle c_k, [l'_k, u'_k]\rangle\rangle$, then there exist $b_1, \ldots, b_k$ such that $\langle\langle c_1, [b_1, b_1]\rangle, \ldots, \langle c_k, [b_k, b_k]\rangle\rangle$ is also a τ-repair of $\mathcal{X}$.*

Proof. All three levels of controllability require that for *all* situations generated by the contingent constraints, the STNU is controllable. Hence, tighter bounds on the contingents can only make an STNU more controllable. □

Proposition 1. *Problems* S-REPAIR*,* W-REPAIR *and* D-REPAIR *are solvable in polynomial time.*

Proof. Lemma 1 shows that for all three levels of controllability, the repair problem has a solution if and only if it has one in which all contingent constraints are repaired to singleton intervals. Moreover, for a tightening ρ of $\mathcal{X} = \langle V, E, C\rangle$ in which all the contingent constraints of $\mathcal{X}$ are repaired to singleton intervals, the STNU $\mathcal{X} \oplus \rho$ is now an STN, since contingent durations are known (i.e., fixed).

It follows that we only have to search a value $\alpha_c \in [l, u]$ for each contingent $c = v_i \xrightarrow{[l,u]} v_j \in C$, such that the STN obtained from $\mathcal{X}$ by replacing each such c by the *requirement* $v_i \xrightarrow{[\alpha_c, \alpha_c]} v_j$, is satisfiable. This can clearly be done by solving a system of linear equations, as illustrated on Figure 2. □



$$\begin{cases} v_1 - v_0 \leqslant \alpha_1; v_1 - v_0 \geqslant \alpha_1 \\ v_3 - v_2 \leqslant \alpha_2; v_3 - v_2 \geqslant \alpha_2 \\ v_3 - v_0 \geqslant 25; v_3 - v_0 \leqslant 30 \\ v_2 - v_1 \geqslant 5; v_2 - v_2 \leqslant 15 \\ 5 \leqslant \alpha_1 \leqslant 12; 10 \leqslant \alpha_1 \leqslant 15 \end{cases}$$

Figure 2: Searching for a valid tightening of contingent constraints $\rho = \langle\langle c_1, [\alpha_1, \alpha_1]\rangle, \ldots, \langle c_2, [\alpha_2, \alpha_2]\rangle\rangle$ of the STNU of Figure 1 is equivalent to solving consistency of a polynomial size linear system where each α_i are variables.

5.2 Optimization Repairs for Strong Controllability

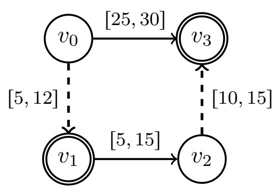
In this section, we will provide a new and polynomial algorithm for repairing a non-SC STNU that can be adapted for the partial and k-budget repair problems. For the k-constraint repair problem, we will reduce it to the SUBSET-SUM problem to prove that it is NP complete.

In [19], the authors prove that checking Strong Controllability can be reduced to solving a system of linear equations over the requirement constraints. In fact, any STNU can be encoded the same way as for STNs (see Figure 2) by replacing in the equations each uncontrollable timepoint $v_j \in \mathcal{V}_u$ by the relation between a controllable timepoint it relates to and the uncontrollable duration of the contingent constraint between them. The most obvious case is the one of a requirement constraint $v_i \xrightarrow{[l,u]} v_j$ where v_j is uncontrollable and relates to $v_k \xrightarrow{[l',u']} v_j$: in that case we write $v_j - v_i \in [l, u]$ as follows: $\begin{cases} (v_k + \omega_j) - v_i \geqslant l \\ (v_k + \omega_j) - v_i \leqslant u \end{cases}$

There exist three cases where an uncontrollable timepoint is involved: the one we showed where $v_j \in \mathcal{V}_u$, the one where $v_i \in \mathcal{V}_u$, and where $v_i \in \mathcal{V}_u$ and $v_j \in \mathcal{V}_u$. Nonetheless, the methodology remains the same, leading to a system of linear equations of polynomial size.

One can see that a schedule that satisfies such a system, whatever the duration of $\omega = \{\omega_1, \ldots, \omega_n\}$, is a 'strong' schedule that satisfies the SC Definition 6. Hence, if a schedule exists where for each linear equation ω takes the worst possible value, then such a schedule is a strong schedule. The worst possible value for ω is when the contingents take their lower/upper bound value. Thus, finding a solution is done in polynomial time. Figure 3 shows the encoding of checking SC with the STNU of Figure 1. First, we show with ω_1 and ω_2, then with the worst possible value for ω_1 and ω_2 in each linear equation. Formally, we have the following.

Proposition 2 ([19]). *Given an STNU $\mathcal{X}$, one can build*



$$\begin{cases}(v_2+\omega_2)-v_0 \geqslant 25; (v_2+\omega_2)-v_0 \leqslant 30\\ v_2-(v_0+\omega_1) \geqslant 5; v_2-(v_0+\omega_1) \leqslant 15\end{cases}$$
$$\begin{cases}(v_2+10)-v_0 \geqslant 25; (v_2+15)-v_0 \leqslant 30\\ v_2-(v_0+12) \geqslant 5; v_2-(v_0+5) \leqslant 15\end{cases}$$
$$\begin{cases}(v_2+L_1)-v_0 \geqslant 25; (v_2+U_1)-v_0 \leqslant 30\\ v_2-(v_0+15) \geqslant 5; v_2-(v_0+10) \leqslant 15\\ 5 \leqslant L_1 \leqslant U_1 \leqslant 15\end{cases}$$

Figure 3: Reduction resulting from checking Strong Controllability. The two requirements to encode illustrate the first two cases of uncontrollable timepoints: $v_i \in \mathcal{V}_u$ with v_1, and $v_j \in \mathcal{V}_u$ with v_3. We highlight the worst-case scenario for each linear equation with the bounds' value. Then, the encoding to the partial-Strong-repair problem by replacing M_1's values (bounds) with proper variables L_1 and U_1.

in polynomial time a linear program over one variable per requirement constraint of $\mathcal{X}$, such that the solutions of this program are in one-to-one correspondence with the schedules which ensure that $\mathcal{X}$ is strongly controllable.

Proposition 3. *Problems* PARTIAL-S-REPAIR *and* k-BUDGET-S-REPAIR *are solvable in polynomial time.*

Proof. Given Proposition 2, it is easy to build a linear program of polynomial size which decides partial repairability, and k-budget repairability:

- for partial repairability, we replace in the linear program of Proposition 2, each numeric value corresponding to a lower (resp. upper) bound of a repairable contingent constraint $c = v \xrightarrow{[l,u]} v'$, by a new variable L (resp. U), and we add linear constraints $l \leqslant L \leqslant U \leqslant u$; it is then easy to see that the solutions of this linear program are in one-to-one correspondence with the repairs (of each c to $[L,U]$) and the corresponding strong schedules of the STNU. Figure 3 illustrates the encoding for the partial-Strong-repair of Figure 1 assuming only M_1 can be reduced;
- for k-budget repairability, we similarly replace the numeric values of all contingent constraints, again enforce $l \leqslant L \leqslant U \leqslant u$ for all of them, and additionally enforce the linear constraint:

$$\sum_{v \xrightarrow{[l,u]} v' \in C} (L-l+u-U) \leqslant k$$

□

Proposition 4. *Problem* k-CONSTRAINT-S-REPAIR *is* NP*-complete.*

Proof. For membership in NP, one simply need to guess a tightening ρ, consisting of the new bounds l', u' for at most k contingent links, then check that $\mathcal{X} \oplus \rho$ is strongly controllable (in polynomial time using Proposition 2). Since STNUs consist of linear constraints only, it is easy to see that one can restrict to guessing polynomial-size numbers (in the size of the numbers appearing in the STNU).
For hardness, we give a reduction from SUBSET-SUM, which is defined as follows and known to be NP-complete [9, Sec. A3.2].

- *Input:* A multiset of (strictly) positive integers $\{\{n_1,\dots,n_\ell\}\}$ and an integer N
- *Question:* Is there $I \subseteq \{1,\dots,\ell\}$ satisfying $N = \sum_{i\in I} n_i$?

Given an instance $S = \langle\{\{n_1,\dots,n_\ell\}\},N\rangle$, we write $M = \sum_{i=1}^{\ell} n_i$, and we define an STNU $\mathcal{X}^S$ derived from the multiset S as follows (see Figure 4):

- the set of controllable timepoints is $V_c^S = \{v_0,v_1,\dots,v_\ell\}$, with v_0 acting as the reference timepoint, and the set of uncontrollable timepoints is $V_u^S = \{v_1^{\mathsf{y}},\dots,v_\ell^{\mathsf{y}}\} \cup \{v_1^{\mathsf{n}},\dots,v_\ell^{\mathsf{n}}\}$;
- the set of requirement constraints is $E^S = \{e_1^{\mathsf{y}},\dots,e_\ell^{\mathsf{y}}\} \cup \{e_1^{\mathsf{n}},\dots,e_\ell^{\mathsf{n}}\} \cup \{e^N\}$, with $e_i^{\mathsf{y}} = v_i^{\mathsf{y}} \xrightarrow{[0,n_i]} v_i$, $e_i^{\mathsf{n}} = v_i^{\mathsf{n}} \xrightarrow{[0,+\infty]} v_i$ $(i=1,\dots,\ell)$, and $e^N = v_0 \xrightarrow{[M+N,M+N]} v_\ell$;
- the set of contingent constraints is $C^S = \{c_1^{\mathsf{y}},\dots,c_\ell^{\mathsf{y}}\} \cup \{c_1^{\mathsf{n}},\dots,c_\ell^{\mathsf{n}}\}$, with $c_i^{\mathsf{y}} = v_{i-1} \xrightarrow{[0,n_i]} v_i^{\mathsf{y}}$ and $c_i^{\mathsf{n}} = v_{i-1} \xrightarrow{[n_i,2n_i]} v_i^{\mathsf{n}}$ $(i=1,\dots,\ell)$.

Moreover, we define the number of repairable constraints k to be ℓ.
Clearly, $\mathcal{X}^S$ can be constructed in polynomial time given S. We now claim that S is a positive instance of SUBSET-SUM if and only if $\mathcal{X}^S$ is k-constraint strongly repairable.
First assume that I is a solution of S, that is, $N = \sum_{i\in I} n_i$ holds. We define the k-constraint tightening ρ^I to be $\langle\langle c_i^{\mathsf{y}},[n_i,n_i]\rangle \mid i \in I\rangle \cdot \langle\langle c_j^{\mathsf{n}},[n_j,n_j]\rangle \mid j \notin I\rangle$ ($\cdot$ denotes concatenation of tuples). Since exactly one constraint per index $i \in \{1,\dots,\ell\}$ is repaired, we indeed have $|\mathrm{Supp}(\rho^I)| \leqslant k$. Finally, we define a set of decisions dec by $dec(v_i) = \sum_{j\leqslant i} n_j + \sum_{j\leqslant i, j\in I} n_j$ for $i = 0,\dots,\ell$ (in particular, $dec(v_0) = 0$). Then it is easy to prove that the schedule δ induced by dec and ω satisfies all the constraints in E^S, so that ρ^I is a k-constraint strong repair of $\mathcal{X}^S$.
Conversely, assume that ρ is a k-constraint strong repair of $\mathcal{X}^S$. We first show that for $i = 1,\dots,\ell$, ρ repairs either c_i^{y} or c_i^{n} to $[n_i,n_i]$. For contradiction, assume first that it repairs neither, and let ω be a situation with $\omega(c_i^{\mathsf{y}}) = 0$ and $\omega(c_i^{\mathsf{n}}) = 2n_i$; then in order to satisfy constraints e_i^{y} and e_i^{n},

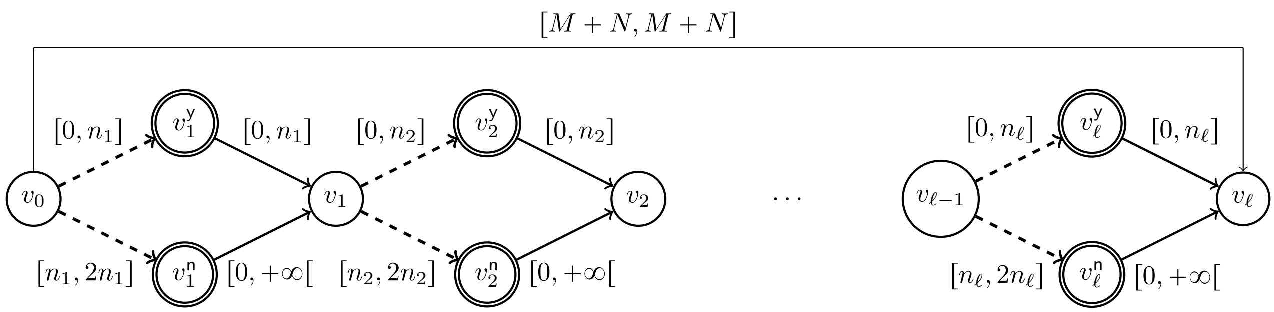


Figure 4: The STNU $\mathcal{X}_S$ built from an instance S of SUBSET-SUM.

a schedule δ must satisfy

$$\delta(v_{i-1}) + 2n_i = \delta(v_{i-1}) + \omega(c_i^{\mathsf{n}}) \leqslant \delta(v_i) \leqslant \delta(v_{i-1}) + \omega(c_i^{\mathsf{y}}) + n_i = \delta(v_{i-1}) + n_i,$$

contradicting $n_i > 0$. Hence ρ repairs at least one of c_i^{y} and c_i^{n} for each $i = 1, \dots, \ell$, and given the budget of $k = \ell$ constraints, it must repair exactly one for each i.
First consider the case when ρ repairs c_i^{y}, and let δ be a strong schedule in the repaired STNU. Then δ is valid in particular for a situation ω with $\omega(c_i^{\mathsf{n}}) = 2n_i$, so that we have

$$\delta(v_{i-1}) + 2n_i = \delta(v_{i-1}) + \omega(c_i^{\mathsf{n}}) \leqslant \delta(v_i) \leqslant \delta(v_{i-1}) + \omega(c_i^{\mathsf{y}}) + n_i,$$

which entails $\omega(c_i^{\mathsf{y}}) = n_i$ and (hence) $\delta(v_i) = \delta(v_{i-1}) + 2n_i$. Dually, if ρ repairs c_i^{n} and δ is a strong schedule in the repaired STNU, then considering a situation ω with $\omega(c_i^{\mathsf{y}}) = 0$, we have

$$\delta(v_{i-1}) + n_i = \delta(v_{i-1}) + \omega(c_i^{\mathsf{y}}) + n_i \geqslant \delta(v_i) \geqslant \delta(v_{i-1}) + \omega(c_i^{\mathsf{n}}),$$

which entails $\omega(c_i^{\mathsf{n}}) = n_i$ and $\delta(v_i) = \delta(v_{i-1}) + n_i$.
In the end, a strong schedule δ in the repaired STNU satisfies for each $i = 1, \dots, \ell$ either $\delta(v_i) = \delta(v_{i-1}) + n_i$ or $\delta(v_i) = \delta(v_{i-1}) + 2n_i$. Hence it satisfies $\delta(v_\ell) - \delta(v_0) = M + \sum_{i \in I} n_i$ for some subset I of $\{1, \dots, \ell\}$; moreover, it satisfies $\delta(v_\ell) - \delta(v_0) = M + N$ due to constraint e^N, so $\sum_{i \in I} n_i = N$ holds and S is a positive instance of SUBSET-SUM. □

5.3 Optimization Repairs for Dynamic Controllability

This section will show that partial-Dynamic-repair and k-budget Dynamic repair problems are in NP, while the k-constraint Dynamic repair is NP-complete.

Proposition 5. PARTIAL-D-REPAIR *and* k-BUDGET-D-REPAIR *are in* NP.

Proof. The proof is similar to that for Strong repair (Proposition 4): we can guess a repair and then check that it is indeed a repair in polynomial time [13]. □

Proposition 6. *Problem* k-CONSTRAINT-D-REPAIR *is* NP-*complete.*

Proof. Membership in NP follows exactly from the same reasoning as in the proof of Proposition 5. For hardness, we use exactly the same reduction as in the proof of Proposition 4, and we show that $\mathcal{X}^S$ is k-constraint strongly repairable if and only if it is k-constraint dynamically repairable. One direction is obvious, since by definition Strong controllability implies Dynamic controllability.
Conversely, the proof that at least one contingent link per i (and hence exactly one) must be repaired applies verbatim from Proposition 4. Now for $i \in \{1, \dots, \ell\}$, let δ_i be a dynamic schedule in the repaired STNU, for the case when ω satisfies $\omega(c_i^{\mathsf{n}}) = 2n_i$ (resp. $\omega(c_i^{\mathsf{y}}) = 0$) if c_i^{y} (resp. c_i^{n}) has been repaired. Then δ_i is fixed, and the same reasoning as in the proof of Proposition 4 applies, concluding that $\mathcal{X}$ is (k-constraint) strongly repairable. □

5.4 Optimization Repairs for Weak Controllability

This section will show that partial-Weak repair, k-budget Weak repair, and k-constraint Weak repair problems are at least coNP-hard and at most in Σ_2^{P}.

Proposition 7. PARTIAL-W-REPAIR, k-BUDGET-W-REPAIR *and* k-CONSTRAINT-W-REPAIR *are* coNP-*hard.*

Proof. We consider particular cases for which the problem comes down to checking Weak Controllability which is known to be coNP-complete [12]. By setting $R = \varnothing$, and $k = 0$, then no contingent can be repaired and the STNU is repairable if and only if it is weakly controllable. Therefore, all of them are as hard as the problem of checking WC. □

Proposition 8. PARTIAL-W-REPAIR, k-BUDGET-W-REPAIR *and* k-CONSTRAINT-W-REPAIR [1] *are in* Σ_2^{P}.

Proof. It is possible to guess in polynomial time a tightening ρ such that $\mathrm{Supp}(\rho) \subseteq R$ for the partial-Weak repair problem, $\mathrm{Cost}(\rho) = k$ for the k-budget Weak repair

[1] We recently found a proof that k-constraints Weak repair problem is Σ_2^{P}-complete. Indeed, Morris et al. proposed a reduction of the famous 3-color problem to WC checking of STNUs [12]. We can extend their proof to reduce the problem 2-round 3-colorability, which is known to be Σ_2^{P}-complete [1, Theorem 11.4], to k-constraints Weak repair problem. The completeness remains open for the other two repair problems.

problem, and $|\mathrm{Supp}(\rho)| = k$ for the k-constraint Weak repair problem. Checking that $\mathcal{X} \oplus \rho$ is weakly controllable can be verified by a coNP oracle. Hence, we get the membership to Σ_2^{P}. □

6 The Multi-agent Case

In this section, we discuss the complexity of the repair problem for more than one agent. For instance, the repair problem for MISTNU is more than relevant and determining its complexity is more challenging [16].

There exist two ways of repairing STNUs in a multi-agent system. One can try to centralize the repair problem using some global function that merges the local repair problem of each agent into a big problem. This is possible when formulating the problem using a first-order formula or a system of equations to solve as done in [5, 16]. In that particular case, where the repair problem is solved in a centralized way, its complexity remains the same as repairing only one STNU.

The real challenge arises when solving the repair problem distributedly is required. More (external) criteria exist that may influence the hardness of the repair problem, related to the information that is exchanged and the size of the messages, which can grow exponentially, or to communication failure or possible delays (of uncertain duration), zone of communication (dependent on the application) that can impact when information will be known; related to the nature of the agent (cooperative or selfish); or related to the global architecture and policy of the agents: pre-existing hierarchy, or on the contrary fairness issues; etc. Therefore, depending on the scenario, one repair may be harder than another and, thus, does not belong to the same class. Nonetheless, this would require a thorough evaluation of the distributed repair problem.

7 Conclusion

In this paper, we studied the complexity of the repair problem for STNU by introducing four definitions of this problem. We evaluated each of them regarding the semantics of each controllability level. At the same time, we proved their completeness for most of them. The problem remains open for some of them.

Future works will study the complexity of the repair problem for the higher class of Temporal Network with Uncertainty not only for the single agent case, but also for the multi-agent case.

References

[1] Miklós Ajtai, Ronald Fagin, and Larry J. Stockmeyer. The closure of monadic NP. *J. Comput. Syst. Sci.*, 60(3):660–716, 2000.

[2] Shyan Akmal, Savana Ammons, Hemeng Li, and James C Boerkoel Jr. Quantifying degrees of controllability in temporal networks with uncertainty. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 22–30, 2019.

[3] Shyan Akmal, Savana Ammons, Hemeng Li, Michael Gao, Lindsay Popowski, and James C. Boerkoel. Quantifying controllability in temporal networks with uncertainty. *Artificial Intelligence*, 289:103384, 2020.

[4] Arthur Bit-Monnot, Malik Ghallab, Félix Ingrand, and David E. Smith. FAPE: a constraint-based planner for generative and hierarchical temporal planning. *CoRR*, abs/2010.13121, 2020.

[5] 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 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 930–938. IOS Press, 2016.

[6] Amedeo Cesta, Angelo Oddi, and Stephen F. Smith. A constraint-based method for project scheduling with time windows. *J. Heuristics*, 8(1):109–136, 2002.

[7] Jing Cui, Peng Yu, Cheng Fang, Patrik Haslum, and Brian Charles Williams. Optimising bounds in simple temporal networks with uncertainty under dynamic controllability constraints. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015*, pages 52–60. AAAI Press, 2015.

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

[9] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[10] Luke Hunsberger and Roberto Posenato. Speeding Up the RUL^{-} Dynamic-Controllability-Checking Algorithm for Simple Temporal Networks with Uncertainty. In *36th AAAI Conf. on Artificial Intelligence 2022*, pages 9776–9785, 2022. DOI: 10.1609/aaai.v36i9.21213.

[11] 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.

[12] Paul H. Morris and Nicola Muscettola. Managing temporal uncertainty through waypoint controllability. In *Proceedings of the Sixteenth International Joint Conference on Artificial*

Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages, pages 1253–1258. Morgan Kaufmann, 1999.

[13] 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*, pages 494–502. Morgan Kaufmann, 2001.

[14] Aïdin Sumic and Thierry Vidal. A more efficient and informed algorithm to check weak controllability of simple temporal networks with uncertainty. In *18èmes Journées d'Intelligence Artificielle Fondamentale et 19èmes Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes, JIAF-JFPDA 2024, La Rochelle, France, July 1-3, 2024*, pages 159–169, 2024.

[15] Ajdin Sumic, Alessandro Cimatti, Andrea Micheli, and Thierry Vidal. Smt-based repair of disjunctive temporal networks with uncertainty: Strong and weak controllability. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research - 21st International Conference, CPAIOR 2024, Uppsala, Sweden, May 28-31, 2024, Proceedings, Part II*, volume 14743 of *Lecture Notes in Computer Science*, pages 176–192. Springer, 2024.

[16] Ajdin Sumic, Thierry Vidal, Andrea Micheli, and Alessandro Cimatti. Introducing interdependent simple temporal networks with uncertainty for multi-agent temporal planning. In *31st International Symposium on Temporal Representation and Reasoning, TIME 2024, October 28-30, 2024, Montpellier, France*, volume 318 of *LIPIcs*, pages 13:1–13:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[17] Alessandro Valentini, Andrea Micheli, and Alessandro Cimatti. Temporal planning with intermediate conditions and effects. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9975–9982. AAAI Press, 2020.

[18] Gérard Verfaillie, Cédric Pralet, and Michel Lemaître. How to model planning and scheduling problems using constraint networks on timelines. *Knowl. Eng. Rev.*, 25(3):319–336, 2010.

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

[20] Benjamin W. Wah and Dong Xin. Optimization of bounds in temporal flexible planning with dynamic controllability. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), 15-17 November 2004, Boca Raton, FL, USA*, pages 40–48. IEEE Computer Society, 2004.