Introducing Interdependent Simple Temporal Networks with Uncertainty for Multi-Agent Temporal Planning

Ajdin Sumic
Technological University of Tarbes, France

Thierry Vidal
Technological University of Tarbes, France

Andrea Micheli
Fundazione Bruno Kessler, Trento, Italy

Alessandro Cimatti
Fundazione Bruno Kessler, Trento, Italy

Abstract

Simple Temporal Networks with Uncertainty are a powerful and widely used formalism for representing and reasoning over convex temporal constraints in the presence of uncertainty called *contingent* constraints. Since their introduction, they have been used in planning and scheduling applications to model situations where the scheduling agent does not control some activity durations or event timings. What needs to be checked is then the *controllability* of the network, i.e., that there is a valid execution strategy whatever the values of the contingents. This paper considers a new type of multi-agent extension, where, as opposed to previous works, each agent manages its own separate STNU, and the control over activity durations is *shared* among the agents: what is called here a *contract* is a mutual constraint controllable for some agent and contingent for others. We will propose a semantically enriched version of STNUs that will be composed into a global *Multi-agent Interdependent STNUs* model. Then, controllability issues will be revisited, and we will focus on the repair problem, i.e., how to regain failed controllability by shrinking some of the shared contract durations, here in a centralized manner.

2012 ACM Subject Classification Computing methodologies

Keywords and phrases Temporal constraints satisfaction, uncertainty, STNU, Controllability checking, Explainable inconsistency, Multi-agent planning

Digital Object Identifier 10.4230/LIPIcs.TIME.2024.13

1 Introduction

In many domains, such as planning and scheduling, systems diagnosis and control, etc, one needs to explicitly represent activities that may or must not overlap in time, last over some duration, and synchronize with timestamped expected events [4, 16, 15, 1]. The most commonly used model is Temporal Constraint Networks (TCN) [5]: nodes are time-points, and edges express sets of possible durations relating them. A key issue is the ability to check the temporal satisfiability of the plan/system/process through the *consistency* of the TCN. The simplest class of TCN, called the Simple Temporal Network (STN), arises when they have only binary constraints with convex intervals of values. Consistency checking is made through polynomial-time propagation algorithms.

A well-known extension of STNs that handles uncertainties, called STNU (Simple Temporal Network with Uncertainty), has been proposed by [17]. An STNU contains uncertain (*contingent*) durations between time-points, which means the effective duration is not under the control of the agent executing the plan, which is useful for addressing realistic dynamic and stochastic domains.

31st International Symposium on Temporal Representation and Reasoning (TIME 2024).
Editors: Pietro Sala, Michael Sioutis, and Fusheng Wang; Article No. 13; pp. 13:1–13:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In STNUs, temporal consistency has been redefined as *controllability*: an STNU is controllable if a strategy exists for executing the schedule, whatever values the contingent durations take. In [17], the authors introduce three levels of controllability: *Weak Controllability* (WC), *Dynamic Controllability* (DC), and *Strong Controllability* (SC). These levels depend on how and when the uncertainties are resolved, i.e., the actual durations are observed/known. Different checking approaches have been proposed, widely discussed, and improved [17, 11, 9].

Considering Multi-agent agents interacting in a common environment, each with its own set of temporal activities, have been studied but only for multiple STNs [7] or for a global multi-agent STNU, all agents considering the same kind of contingent durations set by Nature [3] which hence cannot be modified in any way.

But there has been no work addressing the case where temporal coordination is needed due to the uncertainty for one agent coming from decisions made by another one: the duration of a shared activity is controllable (hence *flexible*) for say agent A_1 and contingent (only observed) for agent A_2.

After exposing the most relevant related work in section 2, the paper focuses on our main contributions: (1) it revisits in section 3 the STNU model by formally defining the execution and observation semantics; (2) it proposes in section 4 a new global model called *Multi-agent Interdependent STNUs* (MISTNU), in which one can represent activity durations whose status differs among distinct agents, and it characterizes the three levels of controllability at the MISTNU level. Last, section 5 focuses on the *repair* problem, i.e., how to tighten the negotiable contingent constraints to recover controllability when checking has failed, proposing a first approach and experiments through a SMT-based encoding to synthesize valid repairs for Weak Controllability, before concluding our contributions with a few prospects.

2 Related work

In the literature, some works have tackled the problem of Multi-agent Simple Temporal Networks (MaSTN) in a centralized manner by decoupling a global STN into sub-networks, to distribute the control of a temporal plan among a group of agents in real-time execution scenarios [7, 8]. A fully distributed approach with the notion of privacy between agents is given in [2]. Still, this approach is incomplete in the sense that agents must agree in advance on some fixed durations, which prevents more dynamic solutions from being found. STNUs have received less attention in multi-agent settings, except for the MaSTNU model [3], which proposes a centralized approach to manage a multi-agent plan. This work decouples an STNU into sub-networks, ensuring all of them are dynamically controllable, using a Mixed Integer Linear Programming approach. A more decentralized approach that assumes a central agent that generates candidate decoupling solutions using the limited shared information of the agents is proposed in [18]. The authors ensure each agent independently tests the candidate on its local network and reports conflicts to guide the decoupling solution generation process until all networks are DC. However, the proposed decoupling algorithm may still prune some solutions. Nevertheless, both assume exogenous contingent constraints, i.e., uncertainties coming from outside the system and contingent for everyone.

To the best of our knowledge, no work in the literature has tackled the problem of a multi-agent system with non-exogenous durations (contingents because they are controlled by another agent). Contrary to previous approaches, this problem requires redefining and extending the expressiveness of STNUs. It also requires, as it will be explained, checking if the networks provided at the planning step are temporally well-formed, another issue that was not considered in previous works.

3 Definitions with Execution and Observation semantics

3.1 A revisited model for the single agent case

A Simple Temporal Network (*STN*) is a pair, (V, E), where V is a set of time-points v_i representing event occurrence times, and E a set of temporal constraints between these time-points, in the form of convex intervals of possible durations [5], in the form $v_j - v_i \in [l_{ij}, u_{ij}]$, with lower bounds $l_{ij} \in \mathbb{R} \cup \{-\infty\}$ and upper bounds $u_{ij} \in \mathbb{R} \cup \{+\infty\}$. Interestingly enough, this model encompasses the qualitative precedence constraint, since v_i *precedes* v_j, noted $v_i \preceq v_j$, iff $l_{ij} \geqslant 0$. A reference time-point v_0 is usually added to V, which is the "origin of time", depending on the application (might be, e.g., the current day at 0:00). The goal is to assign values to time-points such that all constraints are satisfied, i.e., to assign a value to each constraint in its interval domain.

An STN with Uncertainty (*STNU*) is an extension in which one distinguishes a subset of constraints whose values are parameters that cannot be assigned but will be observed [17].

As for the global planning/execution framework, we first recall that for a single agent, one usually reasons upon two phases: plan generation and execution. Considering specific constraints (resources, time, uncertainties) often requires an additional constraint satisfaction phase to validate the generated plan. Here, we focus on the problem of checking the satisfiability of a plan under temporal uncertainty. So we start the definition of our framework with a *planning*, a *validation*, and an *execution* phase.

▶ **Definition 1** (STNU). *An STNU is a tuple (V, E, C) such that:*

- *V is a set of time-points $\{v_0, v_1, \ldots, v_n\}$, partitioned into controllable (V_c) and uncontrollable (V_u) with v_0 the reference time-point such that $\forall i, v_0 \preceq v_i$, $v_0 \in V_c$;*
- *E is a set of requirement constraints $\{e_1, \ldots, e_{|E|}\}$, where each e_k relates two time-points $e_k = v_j - v_i \in [l_{ij}, u_{ij}]$ with, $v_i, v_j \in V$.*
- *C is a set of contingent constraints $\{c_1, \ldots, c_{|C|}\}$, where each c_k relates two time-points $c_k = v_j - v_i \in [l_{ij}, u_{ij}]$ with, $v_i \in V_c$, $v_j \in V_u$, and we have $v_i \preceq v_j$[1]. We will denote $end(c_k)$ as v_j.*

▶ **Definition 2** (Schedule). *A **schedule** δ of an STNU $\mathcal{X} = (V, E, C)$ is a mapping δ from all the controllable time-points to real values where: $\delta = \{\delta(v_1), \ldots, \delta(v_{|V_c|})\}$ with $\forall i, v_i \in V_c$, $\delta : v_i \to \mathbb{R}$*

▶ **Definition 3** (Situation and Projection). *Given an STNU $\mathcal{X} = (V, E, C)$, the **situations** of $\mathcal{X}$ is a set of tuples Ω defined as the cartesian product of contingent domains:*

$$\Omega = \bigtimes_{c \in C} [l_c, u_c]$$

*A **situation** $\omega \in \Omega$ is composed of values noted $\omega_k \in [l_{ij}, u_{ij}]$ for each each $c_k = [l_{ij}, u_{ij}] \in C$. A **projection** $\mathcal{X}_\omega = (V, E \cup C_\omega)$ of $\mathcal{X}$ is an STN where C_ω is obtained by replacing each c_k in C by $c'_k = v_j - v_i \in [\omega_k, \omega_k]$. A schedule δ_ω is a **solution** of $\mathcal{X}_\omega$ if it satisfies all the constraints in $\mathcal{X}_\omega$.*

[1] Since here contingent durations are semantically linked to activities owned by some agent, with a start and end time-points, it is not possible to have a contingent duration between two unordered time-points.

Intuitively, the set of situations defines the space of uncertainty, i.e., the possible values of contingent constraints; a projection substitutes all contingent links with a singleton, forcing its duration to the value appearing in ω. Now, a network shall be deemed controllable if it is possible to schedule the controllable time points to satisfy all requirement constraints in any possible projection. However, that depends on how and when the contingent durations are observed/known by the execution supervisor. Then, to reach a semantically sound definition of the controllability properties, we need to express not only at which time a controllable time-point (resp. a contingent duration) is executed by the agent (resp. set by the owner) but also at which time that value is decided (resp. observed/known) by the execution controller in charge of the agent plan execution.

▶ **Definition 4** (Decisions and Observations). *$\forall v_i \in V_c$, $dec(v_i)$ is the time-point at which $\delta(v_i)$ is **decided** by the execution controller.*
*$\forall \omega_k \in C$, $obs(\omega_k)$ is the time-point at which ω_k is **observed** by the execution controller.*

▶ **Definition 5** (Weak Controllability (WC)). *An STNU $\mathcal{X}$ is **weakly controllable** iff $\forall \omega \in \Omega, \exists \delta_\omega$ such that 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: $\forall v_i \in V_c$, $dec(v_i) \preceq v_i$

▶ **Definition 6** (Strong Controllability (SC)). *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: $\forall v_i \in V_c$, $dec(v_i) = v_0$, and the observations are free: possibly no observation ($\forall \omega_k \in \omega$, $obs(\omega_k) = \varnothing$) or observations during execution that will just update the bounds of the constraints in the network.

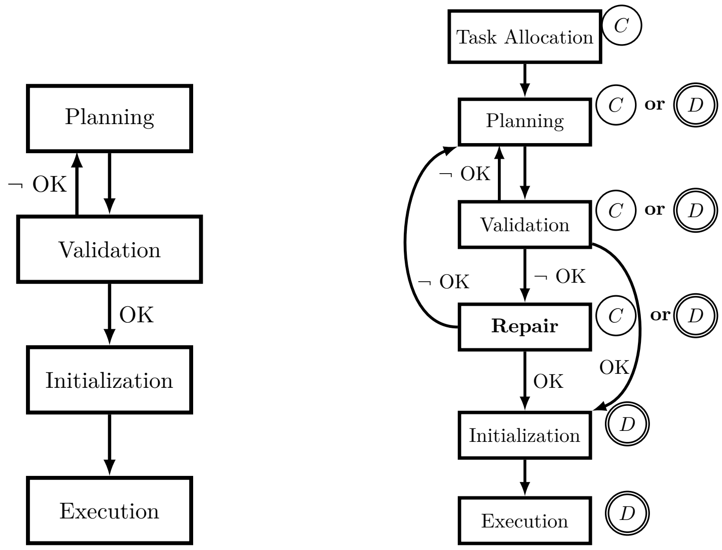
In other words, WC assumes that values of contingent durations will be known **after** plan *validation*, but **before** the *execution* starts. Without any loss of generality, we will consider that all values are set at once exactly at the beginning of the execution: we call this process the *initialization* phase. Then, the schedule can be assigned at the beginning (fixed schedule) or during execution (flexible schedule). For SC, values of contingent durations may be known (or not) at any time since one demands a fixed schedule, which must be set before execution starts, for instance, because users or other agents need to know the precise timing in advance. So, that schedule must be *conformant* to any possible contingent values. Thus, the *initialization* phase will be devoted to schedule assignment.

▶ **Definition 7** (Dynamic Controllability (DC)). *An STNU $\mathcal{X}$ is **Dynamically controllable** iff it is Weakly controllable and $\forall v_i \in V_c, \forall \omega, \omega' \in \Omega$, $\omega^{\preceq v_i} = \omega'^{\preceq v_i} \implies \delta_\omega(v_i) = \delta'_\omega(v_i)$ where $\omega^{\preceq v} = \{\omega_k \in \omega \text{ s.t. } obs(\omega_k) \preceq dec(v)\}$ 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 $\forall v_i \in V_c$, $dec(v_i) = v_i$

In other words, DC assumes that values of contingent durations will be known **during** *execution*, and exactly at the time of occurrence of the ending time-point of the contingent constraint. The schedule is also assigned during execution (flexible schedule), deciding the time of activation of some activity only when all preceding time-points have occurred. Hence, there is no *initialization* phase.

▶ **Example.** *A medical vehicle must visit several villages to offer free COVID testing to the population. The number of people to show off and, hence, the duration of the stay in each village is uncertain. A valid flexible strategy needs to be designed and checked in advance anyway (planning and validation), knowing that the precise information will be known and* *sent to the agent by all the villages one hour before the route begins (initialization). Thus, the agent will know exactly the durations its activities will take and can start executing the plan, which implies WC. If the agent cannot know the number of people waiting in each village in advance, the duration of their testing activities will be known only when the agent arrives, which implies DC. Let's suppose now a rigid valid strategy is required, with village visiting times fixed in advance (hence at the initialization phase at the latest) and no prior knowledge of the number of people in each village. Then SC must be satisfied.*

As a matter of conclusion for the single agent context, we have designed a general framework to deal with temporal uncertainty in planning, including 4 phases: *planning*, *validation*, *initialization*, and *execution*. There is only one possible backtrack: if the validation phase fails (controllability checking fails), the only thing to do is backtrack to the planning engine to find an alternative plan. We show these steps in Figure 1a.



(a) **(b)**

Figure 1 shows the global framework for a single agent (1a) and multiple agents (1b). Node C and double-circled node D refer to the possibility of that step being either centralized (C) or distributed (D). Please note that the initialization phase only exists for WC and SC as DC implies to decide/observe during execution.

3.2 A new temporal multi-agent framework

For a single agent, the contingent duration assignment is exogenous; hence, it is assumed that there is no way to influence them. Saying that "Nature" will assign those values is a usual way to capture that.

However, in a multi-agent environment, a contingent duration may be decided by another agent. So even though the agent that depends on this contingent cannot decide its value, it might be possible to communicate with the *owner* agent to change the possible values. Intuitively, that *owner* agent will decide the duration but commits to assign it within the lower and upper bounds. Some other *compliant* agents depend on that constraint, which is contingent on their network. The former agent (owner) communicates its commitment (lower and upper bound) to the compliant agents at some point before the agents check the controllability of their networks. We call this kind of commitment a *contract* between the owner and the compliant agents, where a compliant agent has the right to request new bounds as long as it guarantees the satisfaction of both agents, i.e., to ensure each is controllable.

First of all, to get a complete picture, we must recall that in multi-agent planning, there might be a first phase of *task allocation* to distribute the goals to achieve to the agents. That phase is usually centralized or devoted to specialized agents. Then the *planning* may be centralized, the global plan being decoupled into separate agent plans or distributed, each agent building its own plan individually. In both cases anyway, dependencies and synchronizations must be considered, calling for some way to share activities controlled by one agent but which outcome is needed by another.

In the end, it is assumed that *execution* will be launched concurrently by all the agents. But before that, after the planning is completed, there is still the need to *validate* the individual plans through, in our case, temporal controllability checking algorithms. Once again, it can be done by a central agent having a view of all the plans or in a distributed way by each agent.

The way that can be done then depends on the observation and decision semantics that have been introduced in the previous section. First, if the application requires that all schedules must be fixed in advance, that means one needs to consider a common *initialization* phase to fix those schedules, which means all agents must ensure SC. Second, if flexible schedules are allowed, then WC or DC apply. The difference depends on how and when the "owner" agents set and communicate the values of activity durations on which other agents are dependent. If that is done before the execution, they must consider a common *initialization* phase when all agents will decide and exchange the shared activities durations, which is consistent with the definition of WC. If such decisions are to be taken during the execution and communicated as soon as they occur and with no delay, then DC applies.

▶ **Example.** *In a hospital environment, a patient has to follow a path through several services that manage their timetabling separately. The path has to satisfy partially ordered constraints between the different services. Now consider that the durations of activities in each service depend, for instance, on the patient features that will only be assessed at the time each activity begins. Then, if all services require a rigid timetable where each operation has a unique starting time that is fixed in advance and appears in the calendar, then SC applies for all; if flexibility is allowed in the sense that operations start times might be decided on the fly, then all services must account for some global DC. Now, consider that each service does not know in advance how many people will be working that day (due to last-minute staff allocation and potential sick leaves), which affects the duration of the patient care. In that case, a plan must be proven valid in advance without such information, but all services will know and exchange them through a common initialization phase when the day starts, which implies WC.*

Of course, this framework is only relevant in homogeneous multi-agent problems when all agents have the same behavior, which we assume here. If not, the classical controllability checking algorithms will not be applicable, which is not our focus. For instance, when agents aim to satisfy different levels of controllability, it results in sharing the decisions of the contracts at different times (before, online, or never). This semantic is different from the one classical controllability checking algorithms assume, as the uncontrollable duration will either be known before execution (WC), online (DC), or never known (SC). This framework also assumes that the initialization phase is synchronous, i.e., all decisions must be taken by all agents at the same time, without communication nor hierarchy between them; otherwise, the semantics of WC (or SC) will not be met, and what needs to be checked will also be something different that is out of the scope of our current study. Hence, extending the well-defined semantics of WC, SC, and somehow DC to a multi-agent setting aligns with specific and somehow restricted semantics of the behavior of the team of agents.

Then, going back to the *validation* phase, if at least one agent is not controllable, then it is still possible to backtrack to the planning phase to find an alternative. Still, it is possible to negotiate with other agents to change the values of some contracts they control. If the "owner" agent agrees to change the bounds of the contract controls so that both agents are now controllable, the problem is solved without needing a more complex replanning stage. This new phase, either centralized or distributed, is the *repair* phase and may require controllability checking algorithms capable of diagnosing the source of uncontrollability [10]. This new phase may also succeed or fail (no solution exists) with no other choice but to backtrack to the planning phase. Figure 1b synthesizes this new global framework.

4 The MISTNU model

4.1 Definitions

The concept of negotiable contingent constraints arises in a multi-agent context when such a constraint is not controlled by Nature but by one agent of the system. Hence, we slightly modify the definition of an STNU in the form of a *Contracting* STNU (cSTNU) by explicitly considering some constraints as *owned* by the agent and relating the contingent constraints to so-called *contracts*, the bounds and the owner of such contracts being now defined outside the model, to be shared by several agents[2].

▶ **Definition 8** (cSTNU). *A Contracting STNU (cSTNU) is an STNU where links representing contracts are labeled. A cSTNU is a tuple $\mathcal{S} = \langle V, R, W, E, C, O \rangle$ such that:*

- *V is a set of time points, partitioned into V_c and V_u (Definition 1)*
- *R and W are sets of contracts owned (W) and observed (R), such that $R \cap W = \varnothing$*
- *E is a set of requirement links of the form $v_i \xrightarrow{[l,u]} v_j$;*
- *C is a set of labeled contingent links of the form $v_i \overset{p}{\dashrightarrow} v_j$ where $p \in R$.*
- *O is a set of owned contract links of the form either $v_i \xrightarrow{p} v_j$ or $v_i \overset{p}{\dashrightarrow} v_j$, one for each contract $p \in W$.*

In addition, we require that $\forall v_j \in V_u$, there exists a unique labeled contingent link of the form $v_i \overset{p}{\dashrightarrow} v_j$ in $C \cup O$.

One can notice that a contract is a labeled link where an agent may consider its owned contracts (in W) as contingent or requirement constraints, depending on its policy (represented by O). Forcing that constraint to be contingent prevents the agent from shrinking it when running some local propagation algorithm to respect its commitment to others or to retain the contract's maximal flexibility at execution time (i.e., it refuses any reduction to allow other agents to regain control). That shall enable us to tune our global model accordingly in settings where agents are more or less selfish or cooperative. An expressiveness is not allowed by the STNU model. In addition, in a multi-agent scenario, it's important to note that requirement constraints in E are private.

Then, in Definition 9, we define the model of Multi-agent Interdependent Simple Temporal Networks under Uncertainty (MISTNU).

[2] As already mentioned, an exogenous contingent constraint can still be modeled here, being related to a contract without any owner; which will be allowed in the global model.

▶ **Definition 9** (MISTNU). *A MISTNU is a tuple $\mathcal{G} = \langle A, \Sigma, B\rangle$ such that:*
- *A is a set of agents $\{a_1, a_2, \ldots, a_n\}$;*
- *Σ is a set of cSTNUs $\mathcal{S}_a = \langle V_a, R_a, W_a, E_a, C_a, O_a\rangle$, one for each $a \in A$, such that*
  - *$\forall a \in A$, $v_z \in V_a$, where v_z is the mutual reference time point: $\forall v_i \in V_a, v_z \preceq v_i$;*
  - *for every pair of agents $a, b \in A$, $W_a \cap W_b = \varnothing$*
- *B is a map from contracts to bounds $B : \bigcup_{a\in A}(R_a \cup W_a) \to \mathbb{R}^2$.*

A MISTNU is a model $\mathcal{G}$ composed of a set of agents, where each agent has its own cSTNU and might own or read contracts; some of them are shared and might be negotiated. Then, $\mathcal{G}$ is also composed of a map of contracts to bounds B that indicates for every contract p its lower/upper bound denoted as a pair $\langle l_p, u_p\rangle$ with l_p and u_p respectively the lower and upper bound of the interval of possible durations. This allows us to reduce a cSTNU into an STNU by applying B:

▶ **Definition 10** (cSTNU reduction). *Given a cSTNU $\mathcal{S} = \langle V, R, W, E, C, O\rangle$ and a map $B : W \cup R \to \mathbb{R}^2$ giving bounds to contracts, $\mathcal{S}$ can be reduced to an STNU $S^{\mathcal{G}} \doteq \langle V, E', C'\rangle$ with:*
- $E' = E \cup \{v_i \xrightarrow{[l_p,u_p]} v_j \mid v_i \xrightarrow{p} v_j \in O,\ B(p) = \langle l_p, u_p\rangle\}$
- $C' = \{v_i \overset{[l_p,u_p]}{\dashrightarrow} v_j \mid v_i \overset{p}{\dashrightarrow} v_j \in C \cup O,\ B(p) = \langle l_p, u_p\rangle\}$

As a cSTNU can be reduced to an STNU with Definition 10, the definitions of its situations and projections directly come from Definition 3. However, for the global MISTNU model, things are a little more complex. We hence first provide further definitions:
- for any agent a, $P_a = R_a \cup W_a$ is the set of all its contracts;
- $P = \bigcup_a P_a$ gathers the contracts of all the agents; $W = \bigcup_a W_a$ the ones having owners.
- for any cSTNU $\mathcal{S}$, $\sigma(\mathcal{S}, p) = v_i$ s.t. $\exists v_j, v_i \overset{\text{p}}{\dashrightarrow} v_j \in C \cup O$ or $v_i \xrightarrow{p} v_j \in O$ denotes the starting time point of contract p in $\mathcal{S}$.

▶ **Definition 11** (MISTNU situation). *Given a MISTNU $\mathcal{G} = \langle A, \Sigma, B\rangle$, the* ***situations*** *of $\mathcal{G}$ is a set of tuples of reals $\Omega_{\mathcal{G}}$ defined as the cartesion product of:*

$$\Omega_{\mathcal{G}} = \bigtimes_{p\in P} [l_p, u_p].$$

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

▶ **Definition 12** (MISTNU projection). *Given a MISTNU $\mathcal{G} = \langle A, \Sigma, B\rangle$, and a situation ω, the* ***projection*** *$\mathcal{G}^\omega$ is a model $\langle A, \Sigma^\omega, B^\omega\rangle$ where:*
- *B^ω is a map from contracts to fixed values $B^\omega : P \to \mathbb{R}^2$ such that $B^\omega = \{\langle \omega(p), \omega(p)\rangle \mid p \in B\}$*
- *Σ^ω is a set of STNs $\mathcal{X}^\omega_a = \langle V_a, E'_a\rangle$, one per $a \in A$, s.t for $\mathcal{S}_a = \langle V_a, R_a, W_a, E_a, C_a, O_a\rangle$ in Σ:*

$$E'_a = E_a \cup \{v_i \xrightarrow{B^\omega(p)} v_j \mid v_i \xrightarrow{p} v_j \in O_a\} \cup \{v_i \xrightarrow{B^\omega(p)} v_j \mid v_i \overset{p}{\dashrightarrow} v_j \in C_a \cup O_a\}$$

It is important to point out that as the system considers temporal networks created independently, the model must ensure that all the temporal networks in Σ are temporally well-formed. This means that for any contract of the form $v_i \xrightarrow{p} v_j$ or $v_i \overset{\text{p}}{\dashrightarrow} v_j$ shared among a set of agents, the date in time on which its execution starts (v_i) and finishes (v_j) must be the same in each of the temporal networks where the contract is involved. As the contract duration between v_i and v_j is guaranteed to be the same by B, we need to ensure that it is also the case for the start time-point v_i.

▶ **Definition 13** (Temporally well-formed). *A MISTNU $\mathcal{G} = \langle A, \Sigma, B\rangle$ is **temporally well-formed** if for every projection $\omega \in \Omega_{\mathcal{G}}$, for every pair of distinct agents a_1 and a_2 and for every contract $p \in P_1 \cap P_2$, all solutions δ_1 of $\mathcal{X}_1^\omega$ and δ_2 of $\mathcal{X}_2^\omega$ are such that $\delta_1(\sigma(\mathcal{S}_1, p)) = \delta_2(\sigma(\mathcal{S}_2, p))$.*

▶ **Theorem 14.** *Let T be a map from a contract p to its unique predecessor p' or v_z, $T : P \to P \cup \{v_z\}$, such that $\forall p$, there is no sequence of the form $T(T(\ldots T(p))) = p$.*

A MISTNU is well-formed if for every agent $a \in A$, $\forall p \in (W_a \cup R_a)$ we have:

- $T(p) = v_z \iff v_z \xrightarrow{p} v_j \in O_a$ *or* $v_z \overset{p}{\dashrightarrow} v_j \in O_a \cup C_a$
- $T(p) = p' \iff (v_i \xrightarrow{p} v_j \in O_a$ *or* $v_i \overset{p}{\dashrightarrow} v_j \in O_a \cup C_a) \wedge (v_k \xrightarrow{p'} v_i \in O_a$ *or* $v_k \overset{p'}{\dashrightarrow} v_i \in O_a \cup C_a)$

Proof. Let's suppose, for the sake of contradiction, that map T exists and follows the constraints of the theorem. Still, the MISTNU $\mathcal{G}$ is not well-formed, then there exists a projection ω, a pair of agents a_1 and a_2 and a contract p such that $\delta_1(\sigma(\mathcal{S}_1, p)) \neq \delta_2(\sigma(\mathcal{S}_2, p))$ as per Definition 13. But we observed from T that $T(p) = v_z$ or $T(p) = p'$. Therefore, by induction over T, we have:

- the base case where $T(p) = v_z$: obviously we have $\delta_1(\sigma(\mathcal{S}_1, p)) = \delta_2(\sigma(\mathcal{S}_2, p)) = 0$ as v_z is the common reference time-point shared among all agents;
- the inductive case where $T(p) = p'$: we assume that $\delta_1(\sigma(\mathcal{S}_1, p')) = \delta_2(\sigma(\mathcal{S}_2, p')) = k$, then $\delta_1(\sigma(\mathcal{S}_1, p)) = \delta_2(\sigma(\mathcal{S}_2, p)) = k + \omega_{p'}$ because in both agents p is started by the end of p' and p' has the same duration for all agents.

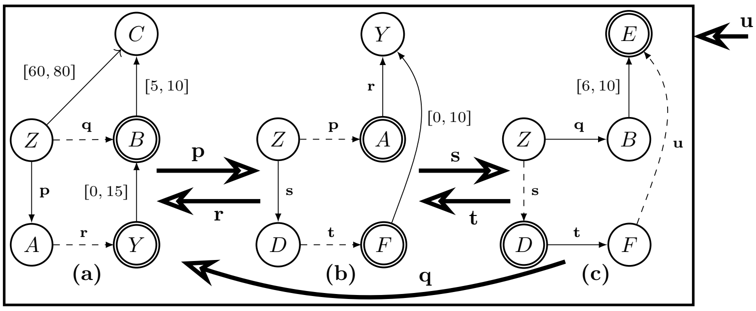
Consequently, it's impossible, from the induction, to have $\delta_1(\sigma(\mathcal{S}_1, p)) \neq \delta_2(\sigma(\mathcal{S}_2, p))$ if $\mathcal{G}$ follows T, and hence $\mathcal{G}$ is guaranteed to be well-formed. ◀

However, Theorem 14 provides a sufficient but not necessary condition for a MISTNU to be well-formed: one could simply require agents to agree on the starting date for a contract or a fixed duration between two contracts. Figure 2 shows an example of a MISTNU with three agents and their networks ensured through the map T to be temporally well-formed with respect to the contracts.

In this example, the contract u comes from an *external* agent, i.e., an agent with which **negotiation is not possible**, which can allow to express several semantic situations: a lack of communication with this agent during the repair, the agent being selfish, etc. In the model, we simply represent it by the contract having no owner, and we must ensure that such contract bounds cannot be shrinked. Nonetheless, such agents shall conform to the semantics and behave in the same way as others: e.g., if WC is considered, that means the duration of the contract u will be shared with its readers during the initialization phase. One can notice that a MISTNU with only one agent is equivalent to a single STNU, with all contracts read by this agent coming from outside the system (not negotiable).

4.2 Controllability

In previous work, the controllability of MaSTNU was defined as having all STNUs controllable [3, 18]. Thus, we define the controllability of MISTNU in the same manner, i.e., to be Dynamically controllable, the system would impose all the networks (cSTNU) to be Dynamically controllable. Then, with Definition 10, we check the controllability of a cSTNU through an STNU reduction, and checking the controllability of STNUs has already been tackled in the literature [12, 9].



(a)

Label	Contract	Repair
p	[20, 25]	[21, 25]
q	[60, 75]	[65, 73]
r	[25, 35]	[33, 35]
s	[30, 35]	[31, 35]
t	[15, 25]	[15, 25]
u	[15, 20]	[15, 20]

(b)

Figure 2 Example of a MISTNU with agents a, b, and c and their networks. Nodes are time-points, uncontrollable ones being double circled, solid/dashed edges are requirement/contingent constraints. The contracts are: p owned by a, r, and s by b, q, t by c, and u by an external agent; communication is represented through larger edges, e.g., a sends p to b. In (b), we show the bounds and repairs of the contracts.

▶ **Definition 15** (Controllability). *Given a MISTNU $\mathcal{G} = \langle A, \Sigma, B \rangle$, we define the **$\tau$-controllability** L_τ of $\mathcal{G}$ with $\tau = \{Weak, Dynamic, Strong\}$ as:*

$$L_\tau \equiv \forall S_a \in \Sigma, S_a^{\mathcal{G}} \text{ is } \tau - controllable.$$

where $S_a^{\mathcal{G}}$ is the STNU obtained from S_a by the cSTNU reduction of Definition 10.

In Section 3.2, we argue that WC in a multi-agent system considers a common initialization phase where all agents will decide and exchange the shared activities durations just before execution. The fact that each agent independently decides the duration of the contracts it owns means that it must ensure that this duration is consistent with the choice over the contract's duration owned by the other agents. In other words, whatever the contract duration an agent decides, it must ensure that there always exists a consistent schedule with this duration, whatever the duration of the contract decided by others it receives, which is related to the definition of Weak controllability. Therefore, the semantics of Weak controllability is imposed to guarantee all possible combinations between the contracts owned by an agent and the ones it does not own to form a consistent STN. Hence, for the validation phase (controllability checking), all the contracts must be considered as contingent constraints to guarantee WC is well-checked. This is not the case for the other two types of controllability: DC implies the agent will decide after observing the decisions of the other agents, and SC supposes the agent will fix a schedule before receiving any information about the contracts owned by the others. Thus, it does not require all the owned contracts to be considered contingents for the validation phase.

5 Defining and solving the repair problem

5.1 The repair problem: definitions

The concept of repair for MISTNU arises when the system is not controllable. We focus on local controllability by finding a tightening (if it exists) of the bounds of an agent's contracts so that local controllability is recovered. In the following, we formalize the repair problem.

▶ **Definition 16** (Repair). *Given a global model $\mathcal{G} = (A, \Sigma, B)$ such that for some agent $a \in A$, $\mathcal{S}_a$ is not τ-controllable with $\tau = \{Weak, Dynamic, Strong\}$. The* ***$L_\tau$-repair problem*** *consists in finding new bounds B' for a global model $\mathcal{G}' = (A, \Sigma, B')$ such that:*

- $\forall p \in W$ [3] *let $\langle l_p, u_p \rangle = B(p)$ and $\langle l'_p, u'_p \rangle = B'(p)$ where $l'_p \geqslant l_p$, $u'_p \leqslant u_p$;*
- *$\mathcal{G}'$ is L_τ-controllable.*

In addition, we are interested in repair solutions that minimize the reduction in the size of the contract bounds, as done in [14] for DTNU and STNU. This intuitively corresponds to minimizing the amount of flexibility removed for each agent concerned by the repair. A unique solution might not exist for this optimal repair, and some general policies might also require fairness in finding an optimal solution, i.e., an optimal repair that equally shrinks the contracts among the agents. In the following, we define the optimal repair and the fair-optimal repair, the latter equally reducing as many contracts as possible by considering the reduction percentage. Somehow, this amounts to finding some optimal equity between the agents. For that, we write $C^2_{|P|}$ as the number of distinct pairs $\langle p_1, p_2 \rangle$ in W.

▶ **Definition 17** (Optimal Repair). *Let $\mathcal{G} = (A, \Sigma, B)$, be a non L_τ controllable MISTNU and let $R_\mathcal{G}$ be the set of all the solutions to the L_τ-repair problem for $\mathcal{G}$. An* ***optimal L_τ-repair*** *for $\mathcal{G}$ is defined as:*

$$\underset{\mathcal{G}' \in R_\mathcal{G}}{\operatorname{argmin}} \left(\sum_{p \to \langle l'_p, u'_p \rangle \in B'} ((l'_p - l_p) + (u_p - u'_p)) \mid \langle l_p, u_p \rangle = B(p) \right)$$

▶ **Definition 18** (Fair-Optimal Repair). *Let $\mathcal{G} = (A, \Sigma, B)$, be a non L_τ-controllable MISTNU and let $R^{opt}_\mathcal{G}$ be the set of all the solutions to the optimal L_τ-repair problem for $\mathcal{G}$. A* ***fair-optimal L_τ-repair*** *for $\mathcal{G}$ is defined as:*

$$\underset{\mathcal{G}' \in R^{opt}_\mathcal{G}}{\operatorname{argmax}} \left(\left| \{ \langle p_1, p_2 \rangle \in C^2_{|P|} \mid \frac{((l'_{p_1} - l_{p_1}) + (u_{p_1} - u'_{p_1}))}{u_{p_1} - l_{p_1}} = \frac{((l'_{p_2} - l_{p_2}) + (u_{p_2} - u'_{p_2}))}{u_{p_2} - l_{p_2}} \} \right| \right)$$

Intuitively, we aim at maximizing the number of pairs of contracts that are shrunk by the same amount. This function selects *among the optimal repairs*, as per Definition 17, solutions in which the reduction of flexibility is shared as much as possible among the contracts. Table 2b shows the WC fair-optimal repair of the MISTNU of Figure 2 with the contracts s and p both being reduced to 20%.

5.2 Encoding of the repair

In this section, we present a centralized encoding for the WC-repair problem of MISTNU based on the encoding for the weak repair of classical STNUs presented in [14]. First, we remind the readers that WC implies all contracts to be contingents. Then, we exploit the convexity of the problem by considering that all combinations of the lower and upper bounds of the contingents are enough to check the controllability of an STNU [17]. This allows us to consider the situations where the duration of a contract is fixed to either its lower-bound or upper-bound for all contract readers. In addition, we define two rational variables for each contract, l^p and u^p, respectively, representing the lower and upper bound of the revised contract p. Please note that a variable is represented with the index as an exponent. Then, we formalize the basic components for the encoding as follows:

[3] This ensures a contract from an external agent, which has no owner, shall not be shrunk (see definition of W in 4.1)

- for each $\mathcal{S}_a$, we define $\vec{X}_a$ as the set of variables $\{v^0, \ldots, v^i\}$ representing the time-points in V_a (agents have disjoint sets of variables).
- $\vec{B}_l = \{l^1, \ldots, l^i\}$ and $\vec{B}_u = \{u^1, \ldots, u^i\}$ the two sets of variables for the lower and upper bound variables of all the contracts.

In addition, for each $\mathcal{S}_a = \langle V_a, R_a, W_a, E_a, C_a, O_a \rangle$, we have the set of projections β_a, one for each possible combination of bounds $\langle l_p, u_p \rangle$ for each contract p of the form $v_i \overset{\text{p}}{\dashrightarrow} v_j$ in $C_a \cup O_a$:

$$\beta_a = \{\omega \mid \omega_p \in \{l_p, u_p\}, v_i \overset{\text{p}}{\dashrightarrow} v_j \in C_a \cup O_a\}.$$

Then, as ω corresponds to a projection, equivalently an STN $\mathcal{X}_\omega = (V_a, E_a \cup C'_a)$, we encode a cSTNU as the conjunction of each projection ω. In addition, from Definition 5, a cSTNU is Weakly controllable if each projection ω has at least one schedule δ_ω. This requires the variables in $\vec{X}_a$ to be unique per projection ω. In this particular case, we denote $\vec{X}_a^\omega = \{v_\omega^0, \ldots, v_\omega^i\}$ the unique set of variables representing the time-points of the projection ω. Then, the MISTNU model can also be encoded as the conjunction of the encoding of each agent's cSTNU ($\mathcal{S}_a$) of the system and the encoding of the contracts' constraints. In the following, we present the encoding of a projection denoted as $\Upsilon_{wc}^{\mathcal{X}_\omega}(\vec{X}_a^\omega)$, the encoding of a cSTNU denoted as $\Upsilon_{wc}^{\mathcal{S}_a}$, the encoding of the contracts denoted $\Psi(\vec{B}_l, \vec{B}_u)$, and the encoding of a MISTNU denoted Υ_{wc}.

$$\Upsilon_{wc}^{\mathcal{X}_\omega}(\vec{X}_a^\omega) = \bigwedge_{t_i \in E'_a} \begin{cases} v_\omega^j - v_\omega^i \in [l^p, u^p] \text{ iff } t_i = v_i \xrightarrow{[\omega_p, \omega_p]} v_j \\ v_\omega^j - v_\omega^i \in [l, u] \text{ iff } t_i = v_i \xrightarrow{[l,u]} v_j \end{cases} \tag{1}$$

$$\Upsilon_{wc}^{\mathcal{S}_a} = \bigwedge_{\omega \in \beta_a} \Upsilon_{wc}^{\mathcal{X}_\omega}(\vec{X}_a^\omega) \tag{2}$$

$$\Psi(\vec{B}_l, \vec{B}_u) = \bigwedge_{p \rightarrow \langle l_p, u_p \rangle \in B} \begin{cases} \{l_p \leqslant l^p \leqslant u^p \leqslant u_p\} \text{ iff } \exists a \mid p \in W_a \\ \{l^p = l_p;\ u^p = u_p\} \text{ iff } \nexists a \mid p \in W_a \end{cases} \tag{3}$$

$$\Upsilon_{wc}(\vec{B}_l, \vec{B}_u) = \left(\bigwedge_{\mathcal{S}_a \in S} \Upsilon_{wc}^{\mathcal{S}_a} \right) \wedge \Psi(\vec{B}_l, \vec{B}_u) \tag{4}$$

Please note that Equation 3 also avoids shrinking contracts from external agents by fixing the variables l^p to l and u^p to u. Then, we solved the optimal WC-repair denoted $\chi_{wc}^{opt}(\vec{B}_l, \vec{B}_u)$ and the fair-optimal WC-repair χ_{wc}^{fair} with a lexicographic optimization process (multi-objective optimization supported by modern Optimization Modulo Theory Solvers [13]) with the optimal WC-repair optimization being the first one to be solved:

$$\chi_{wc}^{opt}(\vec{B}_l, \vec{B}_u) = \text{Minimize} \sum_{p \rightarrow \langle l_p, u_p \rangle \in B} ((l^p - l_p) + (u_p - u^p)) \text{ s.t. } \Upsilon_{wc}(\vec{B}_l, \vec{B}_u) \tag{5}$$

$$\chi_{wc}^{fair}(\vec{B}_l, \vec{B}_u) = \text{Maximize} \left(\left| \{\langle \rho_1, \rho_2 \rangle \in C_{|P|}^2 \mid \rho_1 = \rho_2\} \right| \right) \text{ s.t. } \chi_{wc}^{opt}(\vec{B}_l, \vec{B}_u) \tag{6}$$

where for each distinct pairs $\langle p_1, p_2 \rangle$ in P, we create the variables ρ_1 and ρ_2 such that $\forall p_k \in \{p_1, \ldots, p_{|P|}\}$ we have:

$$\rho_k = \frac{((l^{p_k} - l_{p_k}) + (u_{p_k} - u^{p_k}))}{u_{p_k} - l_{p_k}}$$

5.3 Experiments

In this subsection, we simply show the effectiveness of the proposed approaches for WC repair and evaluate the experimental complexity. We implemented the encoding in Python using the pySMT framework [6]. For our experiments, we use the Z3 solver as the backend. We experimented on a large set of 400 MISTNU limited to four agents based on the T map of Theorem 14.

We randomly and safely generate MISTNUs following these parameters: the size of the networks growing from 10 to 200 nodes; the number of contracts (owned) per agent growing from 1 to 20 (according to the size of the network); the number of edges per network by setting at 3 the number of successors per *divergent* node (only v_z is allowed to have more successors), where a divergent node is a node with more than one successor. We considered the fair-optimal repair for the experiment and ran all the experiments on an Xeon E5-2620 2.10GHz with 3600s/10GB time/memory limits.

We solved more than 60 instances, which was expected as the number of projections (bounds) of all the networks grow exponentially (2^p) [17]. More precisely, the problem of checking Weak Controllability (WC) is expected to be co-NP-complete for a single network. Here, we are solving an optimization problem (repair with fairness), which is harder than the checking problem for multiple networks. This is why we did not solve many instances with a one-hour time limit. But that is only the first approach that calls for improvements.

6 Conclusion

This paper presents a new multi-agent model for temporal problems under uncertainty called MISTNU that considers independent networks where, for an agent network, the execution of some tasks might be controlled by other agents. Hence, an agent can negotiate the duration of such tasks. This paper formally defines the cSTNU model, which is an extension of the STNU model, the MISTNU model, and the problem of checking its controllability. In addition, the paper presents the repair problem for the MISTNU model by shrinking the contracts' bounds and proposes a repair encoding for Weak Controllability. Future work will focus on the repair problem for both SC and DC and on a more efficient one for WC. The proposed MISTNU model works well for homogeneous agents with distributed controllability checking, initialization, and execution. However, a heterogeneous system is more challenging as it implies that agents behave in different ways, which might result in a system with mixed controllability (since observation and execution semantics might be different among the agents). A similar study must be done to get a model capable of managing uncertainties from other agents and the classical uncertainty from the environment (Nature), which has its own semantics.

References

1 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. arXiv:2010.13121.

2 James C. Boerkoel and Edmund H. Durfee. Distributed reasoning for multiagent simple temporal problems. *J. Artif. Intell. Res.*, 47:95–156, 2013. doi:10.1613/JAIR.3840.

3 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 Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *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. doi:10.3233/978-1-61499-672-9-930.

4 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. doi:10.1023/A:1013617802515.

5 Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial intelligence*, 49(1-3):61–95, 1991. doi:10.1016/0004-3702(91)90006-6.

6 Marco Gario and Andrea Micheli. pysmt: a solver-agnostic library for fast prototyping of smt-based algorithms. In *SMT Workshop*, 2015.

7 Luke Hunsberger. Algorithms for a temporal decoupling problem in multi-agent planning. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada*, pages 468–475, 2002. URL: http://www.aaai.org/Library/AAAI/2002/aaai02-071.php.

8 Luke Hunsberger. Distributing the control of a temporal network among multiple agents. In *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings*, pages 899–906, 2003. doi:10.1145/860575.860720.

9 Luke Hunsberger and Roberto Posenato. Speeding up the rul$^-$ dynamic-controllability-checking algorithm for simple temporal networks with uncertainty. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 9776–9785. AAAI Press, 2022. doi:10.1609/AAAI.V36I9.21213.

10 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.

11 Paul H Morris and Nicola Muscettola. Temporal dynamic controllability revisited. In *Aaai*, pages 1193–1198, 2005. URL: http://www.aaai.org/Library/AAAI/2005/aaai05-189.php.

12 Roberto Posenato and Carlo Combi. Adding flexibility to uncertainty: Flexible simple temporal networks with uncertainty (ftnu). *Information Sciences*, 584:784–807, 2022. doi:10.1016/J.INS.2021.10.008.

13 Roberto Sebastiani and Silvia Tomasi. Optimization modulo theories with linear rational costs. *ACM Trans. Comput. Log.*, 16(2):12:1–12:43, 2015. doi:10.1145/2699915.

14 Ajdin Sumic, Alessandro Cimatti, Andrea Micheli, and Thierry Vidal. SMT-based repair of disjunctive temporal networks with uncertainty: Strong and weak controllability. In *Proceedings of the The 21st International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2024)*, 2024.

15 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,*, pages 9975–9982, 2020. doi:10.1609/AAAI.V34I06.6553.

16 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. doi:10.1017/S0269888910000172.

17 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. doi:10.1080/095281399146607.

18 Yuening Zhang and Brian C. Williams. Privacy-preserving algorithm for decoupling of multi-agent plans with uncertainty. In *Proceedings of the Thirty-First International Conference on Automated Planning and Scheduling, ICAPS 2021, Guangzhou, China (virtual), August 2-13, 2021*, pages 426–434. AAAI Press, 2021. URL: https://ojs.aaai.org/index.php/ICAPS/article/view/15988.