State-Temporal Decoupling of Multi-Agent Plans under Limited Communication

Yuening Zhang, Jingkai Chen, Eric Timmons, Marlyse Reeves, Brian Williams
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
zhangyn@mit.edu, jkchen@csail.mit.edu, etimmons@mit.edu, mreeves@mit.edu, williams@mit.edu

Abstract

When a team of agents execute a mission in a distributed fashion, they often communicate with each other to synchronize their progress. However, in situations where communication may be delayed, unavailable or costly, such as when a suite of underwater vehicles is scouting an underwater area in the ocean, pre-coordination is needed beforehand to compensate for limited communication. Previous work proposed decoupling algorithms for Multi-Agent Simple Temporal Network with Uncertainty (MaSTNU) in order to find decoupled execution strategies for the agents, including communication strategy, that satisfy all the inter-agent temporal constraints. However, there is often the coupling between temporal and state constraints, such as the constraint that the vehicles may only communicate with each other when they are within a certain distance. In this paper, we propose using Multi-Agent Qualitative State Plan (MaQSP) that extends MaSTNU to including continuous state constraints in order to model multi-agent plans with coupled state and temporal constraints. We describe a decoupling algorithm for MaQSP using a mixed-integer linear programming (MILP) encoding, which includes a novel path planning algorithm under temporal uncertainty.

Introduction

When multiple agents execute a shared task, the agents often depend on each other, resulting in inter-agent precedence or synchronization constraints. To satisfy those constraints, the agents communicate with each other to synchronize their tasks and update their progress. However, in many cases, the team may operate under limited communication, where communication is not always available and may be delayed or costly. For example, when deploying of a fleet of autonomous underwater vehicles (AUVs) to scout an underwater area in the ocean, communication is mostly unavailable as the AUVs are operating underwater, and has to be planned ahead if communication is required.

Previous work has proposed modeling the multi-agent execution problem as a Multi-Agent Simple Temporal Network (MaSTN) (Hunsberger 2002; Boerkoel Jr and Durfee 2013) or a Multi-Agent Simple Temporal Network with Uncertainty (MaSTNU) (Casanova et al. 2016). They handle limited communication between the team by finding decoupled execution strategies for the agents in the form of a set of local executable networks that depend only on observable information, which is called a temporal decoupling. More specifically, the decoupling for a MaSTN completely removes the need for communication, whereas MaSTNU explicitly models the available communication and its decoupling involves planning for how communication is used to support the mission, resulting in a more flexible coordination strategy. Additionally, MaSTNU also allows the modeling of uncertain durations of activities. While MaSTNU addresses the important problem of temporal coordination under limited communication, it fails to handle the case when there is coupling between temporal constraints and state constraints. In real-life deployments, the AUVs can only communicate when they surface. While AUVs can communicate with each other within a short distance, the ship often has a larger communication range, and can communicate with both of them over a much larger distance. Additionally, there may also exist inter-agent state constraints such as when an AUV finishes its scouting mission, another AUV may want to pick up the scouting mission from where the first AUV left off. Previous work that considers multi-agent coordination with both temporal constraints and continuous state constraints only considers the problem under full observability, modeled as a Qualitative State Plan (QSP) (Léauté and Williams 2005), and its solution is an execution strategy that assumes the existence of a centralized authority that controls all the agents (Fernández-González, Williams, and Karpas 2018; Reeves, Fernández-González, and Williams 2019).

In this paper, we draw insights from the above work, and propose using Multi-Agent Qualitative State Plan (MaQSP) to model both the state and temporal constraints for multi-agent plans under limited communication. With the addition of continuous state constraints, MaQSP allows us to represent the common problem of combined temporal coordination and task planning. We describe a decoupling algorithm for MaQSP by encoding the problem into a mixed-integer linear program (MILP), which depends on a novel path planning algorithm for QSPs under temporal uncertainty assuming first-order dynamics of the agents. Finally, we provide preliminary experiment results on the algorithm.

Motivating Example

Consider a pedagogical example where two AUVs are deployed from the ship on a scouting mission. The vehicle's position is a vector $\langle x, y, d \rangle$, where d is the depth and $d \geq 0$. The initial position is $\langle 0, 0, 5 \rangle$ for both AUVs, and $\langle 0, 0, 0 \rangle$ for the ship. The ship is always above the surface of the water, i.e. $d = 0$ throughout the mission. AUV1 and AUV2 need to take a sample at science site $L1$ and $L2$, respectively. $L1$ region is a rectangular cuboid enclosed by its two corners $[\langle 65, 65, 6 \rangle, \langle 70, 70, 20 \rangle]$ and $L2$ by $[\langle -45, 35, 6 \rangle, \langle -40, 40, 20 \rangle]$. Each of these sampling missions may take any time between 10 to 30 minutes. It is also required that AUV2 must start its sampling mission within 15 minutes after AUV1 has finished its mission.

Notice that since the AUVs are operating underwater and relatively distant from each other, they cannot observe each other's progress. As a result, AUV2 cannot observe when AUV1 has finished its mission, which makes it difficult to satisfy the inter-agent temporal constraint. However, it is possible for the vehicles to communicate their progress by notifying each other upon the occurrence of certain events, though communication is costly and subject to certain constraints. The AUVs can communicate only when they surface. The AUVs can communicate with each other if they are within 30 meters of each other. The AUV and the ship can communicate if they are within 100 meters of each other.

Our problem is to find a coordination strategy so that the AUVs can successfully execute the mission. In this case, because the sampling mission may take any time between 10 to 30 minutes, which cannot be determined beforehand, it becomes necessary for AUV1 to notify AUV2 upon finishing its mission so that AUV2 can react in time to start its mission within 15 minutes. Because the AUVs will be far away from each other in their respective sampling missions, the only possible way to communicate would be to relay the communication from the ship. Additionally, since the AUVs can only communicate when they surface, they must plan for enough time to surface before communication.

Problem Definition

Multi-Agent Qualitative State Plan

We represent the above multi-agent plan by a Multi-Agent Qualitative State Plan (MaQSP), which is an extension of MaSTNU (Casanova et al. 2016) to capture both temporal and state constraints. MaQSP introduces episodes in place of the original temporal constraints, which is a concept from Qualitative State Plan (QSP) (Léauté and Williams 2005). We may also consider MaQSP as an extension of QSP to the multi-agent context. Compared to the typical QSPs, our definition allows the modeling of temporal uncertainty.

Definition 1 (QSP)**.** A *Qualitative State Plan* (QSP) is a tuple $\langle V, X, EP \rangle$, where

- V is a set of events representing designated time points.
- X is a set of continuous state variables.
- EP is a set of *episodes*, where each episode $ep \in EP$ is a tuple $\langle s, t, e, SC, DC \rangle$, in which
  - $s, t \in V$ is the start and end event of the episode.
  - e is a temporal constraint $\langle s, t, lb, ub, ctg \rangle$, where $lb \in \mathbb{R} \cup \{-\infty\}$, $ub \in \mathbb{R} \cup \{+\infty\}$ is the lower bound and upper bound from s to t, i.e. $lb \leq t - s \leq ub$, and ctg is a Boolean indicating if e is a contingent constraint, in which case, $0 \leq lb < ub < \infty$.
  - SC is a set of state constraints scoped on X.
  - DC is a set of delta constraints, where $dc(X_s, X_t) \in DC$ specifies a constraint between the state variables evaluated at the start event and at the end event.

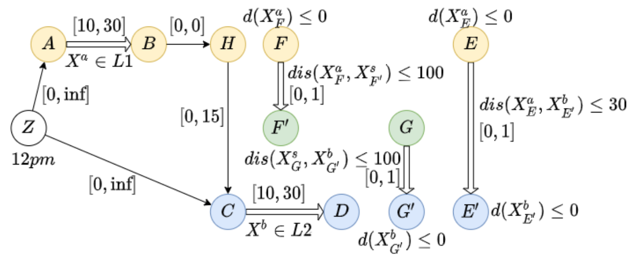

Figure 1: Motivating example in MaQSP

When the ctg flag is set to false, the temporal constraint is a *simple temporal constraint* (Dechter, Meiri, and Pearl 1991), also referred to as a *requirement constraint*, that requires the scheduling of two events to be within certain lower bound and upper bound. When set to true, it is a *simple temporal contingent constraint* (Vidal 1999), or a *contingent constraint* for short, whose end event is an *uncontrollable event* that cannot be directly controlled by the agent, but can be observed when it occurs. A contingent constraint specifies the bound in which the uncontrollable event may occur, and there is a unique contingent constraint for each uncontrollable event. In our example, we can use contingent constraints to express that fact that it may take anytime between 10 to 30 minutes to take a sample, but the duration cannot be determined beforehand. When an episode does not include any state constraints, it is effectively a temporal constraint.

Under the multi-agent context, MaQSP can be considered as a partition of QSP into a set of agents, resulting in a set of agents' local QSPs and additional inter-agent episodes.

Definition 2 (MaQSP)**.** A *Multi-Agent Qualitative State Plan* (MaQSP) is a tuple $\langle N^{\mathcal{A}}, E_X, C_X, v_Z \rangle$, where

- Each $N^a \in N^{\mathcal{A}}$ is the *local state plan* for agent $a \in \mathcal{A}$, which is a QSP $\langle V^a, X^a, EP^a \rangle$, where V^a, X^a, and EP^a are the local events, local state variables, and local episodes for agent a, respectively.
- $E_X \cup C_X$ is a set of external episodes, whose temporal constraint connects the local events of two different agents. *External requirement episodes* E_X and *external contingent episodes* C_X include a requirement and contingent temporal constraint, respectively.
- v_Z is a reference event, an absolute time point proceeding all other events and shared by all agents, such as 12 pm.

Our motivating example can be formulated as a MaQSP shown in Figure 1, where each event is represented by a circle and an episode is represented by an arrow pointing from the start event to the end event, with contingent episodes represented by double arrows. The reference event v_Z is the initial time point 12 pm. Each vehicle's local state plan is highlighted in color, with AUV1 in yellow, AUV2 in blue and the ship in green. For example, AUV1's local state plan consists of local events $\{v_A, v_B, v_E, v_F, v_H\}$ and episode ep_{AB} represents its sampling mission at science site $L1$. The external requirement episode ep_{HC} represents the inter-agent temporal constraint that AUV2 has to start its sampling mission within 15 minutes after AUV1 finishes its mission. The external contingent episodes $C_X = \{ep_{EE'}, ep_{FF'}, ep_{GG'}\}$ are also referred to as the *communication links* that represent available communication between the agents. In this case, we allow AUV1 to communicate to AUV2 once with $ep_{EE'}$, and similarly from AUV1 to the ship $ep_{FF'}$, and the ship to AUV2 $ep_{GG'}$. The timing of communication is unconstrained. For example, $ep_{EE'}$ means that AUV2 can observe the occurrence of event $v_{E'}$ within a delay of 1 min after event v_E is scheduled by AUV1, if they are within 30 meters of each other, AUV1's depth is 0 at event v_E, and AUV2's depth is 0 at event $v_{E'}$.

We characterize three types of state constraints allowed in an episode $ep = \langle v_i, v_j, e, SC, DC\rangle$. Notation-wise, we use X_i to denote the state variables X evaluated at event v_i, and $a(v_i)$ denotes the agent that event v_i belongs to.

- For $sc \in SC_{start}$, sc is satisfied at the start event v_i of the episode. That is, $sc(X_i)$ holds, where $X_i \subseteq X^{a(v_i)}$.
- For $sc \in SC_{end}$, sc is satisfied at the end event v_j of the episode. That is, $sc(X_j)$ holds, where $X_j \subseteq X^{a(v_j)}$.
- For $sc \in SC_{overall}$, sc is satisfied throughout the episode. That is, $\forall T$ s.t. $v_i \leq T \leq v_j$, $sc(X_T)$ holds, where $X_T \subseteq X^{a(v_i)} \cup X^{a(v_j)}$.

In this paper, we consider the following forms of continuous state constraints, where A is a constant matrix, B is a constant vector, and c is a constant:

- $AX \in L$, where L is a convex region approximated by a set of linear inequalities.
- $AX \leq B$, which is a set of linear inequalities.
- $dis(A_1X, A_2X) \leq c$, where A_1 and A_2 has the same size, and A_1X, A_2X usually corresponds to the state variables belonging to different agents.

While the state constraints may take many forms, the above are among the ones typically encountered that also guarantees convexity of our problem. Additionally, we will use $dis(A_1X_s, A_2X_t) \leq c$ to denote distance constraint when it is a delta constraint to differentiate it from an overall state constraint. Note that for distance constraints, since we use a MILP encoding in this paper, we can use L1 distance instead of L2 distance, but we can easily extend it to L2 distance by using MIQCP.

For example, in Figure 1, $X^a \in L1$ for episode ep_{AB} is an overall state constraint that requires AUV1 to stay within science site $L1$ throughout the episode. $d(X^a_E) \leq 0$ for communication link $ep_{EE'}$ is a state constraint to be satisfied at the start event. In our example, $dis(X^a_E, X^b_{E'}) \leq 30$ for $ep_{EE'}$ is a delta constraint that requires the location where AUV1 initiates the communication and the location where AUV2 receives the communication need to be within 30 meters from each other. In this case, we assume the delay is caused by the transmission over media, but the initiation and reception of message is instantaneous. In other cases, it may be reasonable to model a communication link with an overall state constraint $dis(X^a, X^b) \leq 30$ that requires the two vehicles to be within 30 meters of each other throughout the entire communication process, for example, to transmit data. State constraints and delta constraints apply to external requirement episodes too. For example, we may require as a delta constraint that when AUV1 finishes its scouting mission, AUV2 should continue scouting from where AUV1 left off to maintain the consistency of data collected. An example of $SC_{overall}$ may be a tethering constraint, such as when a remotely operated vehicle (ROV) is deployed underwater but is tethered to the ship, it has to stay within a certain distance to the ship throughout the entire mission.

State Temporal Decoupling Problem

Our state temporal decoupling problem for MaQSP is a natural extension of the temporal decoupling problem for MaSTNU (Zhang and Williams 2021).

Definition 3 (State Temporal Decoupling)**.** Given a MaQSP, the set of agents' local state plans $N^{\mathcal{A}}$ forms a *state temporal decoupling* of the MaQSP if:

- (*feasibility*) All local state plans $N^{\mathcal{A}} = \{N^{a_1}, N^{a_2}, \ldots, N^{a_n}\}$ are feasible. That is, there exists a dynamic and valid execution strategy for each local state plan.
- (*validity*) Merging *any* combination of execution strategies for the local state plans $N^{\mathcal{A}}$ yields a solution to the MaQSP, that is, given that the external contingent episodes C_X are satisfied, all the external requirement episodes E_X are also satisfied.

The execution strategy for a local state plan is dynamic as it may need to react on the fly to real-time observations of when the uncontrollable events occur. The execution strategy is valid if the resulting execution satisfies all the temporal and state constraints in the state plan. In this paper, we assume that the evolution of each continuous state variable follows a first-order dynamical model $\dot{x} = v$, where $v \leq v_{max}$ with v_{max} being a fixed maximum change rate.

Definition 4 (Decoupling Problem)**.** The state temporal decoupling problem for MaQSP is a tuple $\langle M, X_0\rangle$, where M is a MaQSP, and $X_0 = \cup_{a\in\mathcal{A}} X^a_0$ specifies the initial state of the agents at the reference event v_Z. The goal is to find a set of *decoupling episodes* for each agent EP^a_d, such that the set of augmented local state plans $N^a_{+\Delta} = \langle V^a, X^a, EP^a \cup EP^a_d\rangle$ for each agent a forms a state temporal decoupling of the MaQSP.

The feasibility condition in Definition 3 requires that the addition of decoupling episodes does not over-constrain any local state plan and makes it infeasible. The validity condition requires that if the local execution strategies satisfy the decoupling episodes, then the external requirement episodes must also be satisfied.

Figure 2 shows an example decoupling for our motivating example, where communication from AUV1 to AUV2 is relayed through the ship and used to support the satisfaction of ep_{HC}. The highlighted red arrows represent the decoupling episodes. For example, ep_{ZF} requires that AUV1

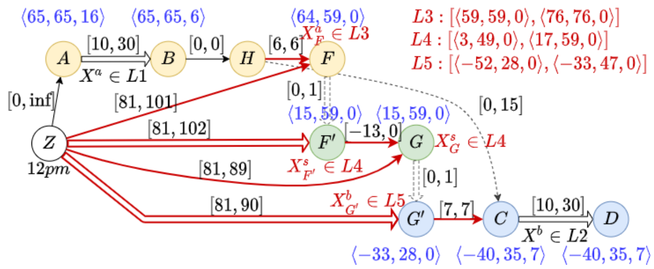

Figure 2: Decoupling solution for motivating example

must execute event v_F between 81 to 101 minutes after v_Z, and it must be in region $L3$ at v_F, where $L3$ is a rectangular cuboid enclosed by its two corners $[\langle 59, 59, 0\rangle, \langle 76, 76, 0\rangle]$. The execution strategy for the ship is that it should schedule v_G as soon as it receives $v_{F'}$ until 89 minutes after v_Z, at which point even if $v_{F'}$ is not received, it should schedule v_G. A feasible trajectory for each vehicle is shown by its $\langle x, y, d\rangle$ position at each event highlighted in blue. Notice that a decoupling solution retains the flexibility for each agent to execute its own state plan, and it only has to enforce the necessary constraints to ensure validity.

Decoupling Algorithm

Our state temporal decoupling algorithm for MaQSP builds on top of the temporal decoupling algorithm for MaSTNU (Casanova et al. 2016) to handle additional continuous state constraints, and follows their use of mixed-integer linear programming (MILP) in order to solve the problem. More specifically, such an encoding involves encoding both the validity condition and the feasibility condition in a single MILP problem, and solving it using off-the-shelf optimization solvers. Intuitively, the validity condition specifies a set of MILP constraints that ensures that the external episodes across agents are decoupled and can be safely removed from the MaQSP without affecting the correctness of the execution result. The feasibility condition specifies a set of MILP constraints that ensures that the local state plans are feasible and can be successfully executed.

Our decoupling algorithm extends the original temporal decoupling algorithm in the following ways: For the validity condition, we use Casanova's encoding to decouple all the external temporal constraints, and add on top of it encoding to decouple all the external state and delta constraints. For the feasibility condition, in the case of MaSTNU, since it only concerns the scheduling problem, its local plan is an instance of simple temporal network with uncertainty (STNU) (Vidal 1999) without any state constraints. Therefore, the feasibility of a STNU is simply its dynamic controllability, which can be encoded as a MILP (Cui and Haslum 2017; Wah and Xin 2007). With additional continuous state constraints, the feasibility of our local state plan becomes a path planning problem under temporal uncertainty. Therefore, we extend the MILP encoding to solve for feasibility of QSP with temporal uncertainty, which is the first to address path planning under temporal uncertainty as we know of.

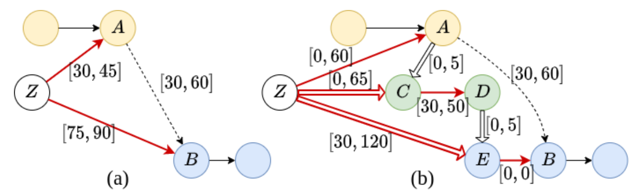

Figure 3: (a) Temporal decoupling without communication (b) Temporal decoupling with communication relay

In the following, we start by describing the temporal validity encoding (Casanova et al. 2016). We then describe the encoding for feasibility of QSP, and the state validity encoding. The final decoupling algorithm puts everything together, which consists of the temporal validity encoding, the state validity encoding, and feasibility constraints for each agent's local QSP.

Temporal Validity Encoding

We start by showing two temporal decoupling examples in Figure 3 to show the intuitive ideas behind the temporal decoupling algorithm. In these examples, the goal is to satisfy the external requirement temporal constraint e_{AB}.

Recall that validity condition requires that by having each agent execute its own local plan, the external requirement constraints are guaranteed to be satisfied. In Figure 3(a), e_{AB} is satisfied by imposing two local temporal constraints, e_{ZA} for AUV1 and e_{ZB} for AUV2. Note that since event v_Z is a reference time point shared by all agents, any constraint connected to it is considered a local constraint. Assuming these local constraints are satisfied, we have $v_B - v_A = (v_B - v_Z) + (v_Z - v_A) = [75, 90] + [-45, -30] = [30, 60]$, which satisfies e_{AB}. Intuitively, we have constrained the execution time window for the start and end events of e_{AB} to be relative to a common reference time point. This is the simplest case of decoupling that requires no communication, first proposed by Hunsberger (Hunsberger 2002).

In Figure 3(b), we additionally have communication links e_{AC} and e_{DE}, similar to our motivating example. In this case, e_{AB} is satisfied by imposing local temporal constraints e_{ZA} for AUV1, e_{ZC} and e_{CD} for the ship, and e_{ZE} and e_{EB} for AUV2. Since e_{AC} and e_{DE} are contingent constraints, we can assume that they are satisfied by nature, and we have $v_B - v_A = (v_B - v_E) + (v_E - v_D) + (v_D - v_C) + (v_C - v_A) = [0, 0] + [0, 5] + [30, 50] + [0, 5] = [30, 60]$, which also satisfies e_{AB}. In this case, not only do we need to satisfy the external requirement constraint e_{AB}, due to the existence of communication links, the agents receiving the communication need to have some expectation of when communication will occur. For example, by constraining e_{ZA}, we are guaranteed that event v_C will definitely occur some time in between $v_C - v_Z = (v_C - v_A) + (v_A - v_Z) = [0, 5] + [0, 60] = [0, 65]$, which is a contingent temporal constraint, since v_C is not controlled by the ship but can only be observed as it occurs.

With the above intuition, the key idea behind finding these decoupling constraints is that the imposed local decoupling constraints need to be tighter or more restrictive than the external requirement temporal constraints to make them redundant, and they should also ensure that the uncontrollable end event of each communication link has a corresponding local contingent constraint that covers all of its possible range of time occurrence. Note that the problem of finding valid temporal decoupling constraints is combinatorial, and hence it is encoded as a MILP summarized below. Readers should refer to (Casanova et al. 2016; Zhang and Williams 2021) for more detail.

Given MaSTNU $\langle N^{\mathcal{A}}, E_X, C_X, v_Z \rangle$, the MILP formulation includes the following variables, where $V = \cup_{a \in \mathcal{A}} V^a$:

(1) Real variables u_{ij} for $v_i, v_j \in V$, with $u_{ii} = 0$.

(2) Boolean variables c_{kj} for $(v_i, v_j, v_k) \in T$, where $T = \{(v_i, v_j, v_k) | e_{ij} \in C_X, v_k \in V^{a(v_j)} \backslash \{v_j\}\}$.

(3) Boolean variables b_{ij} for $(v_i, v_j) \in \overline{E_X}$, where $\overline{E_X} = \{(v_i, v_j) | a(v_i) \neq a(v_j), e_{ij} \notin C_X, e_{ji} \notin C_X\}$.

(4) Boolean variables z_{ijkl} for $(v_i, v_j, v_k, v_l) \in Q$, where $Q = \{(v_i, v_j, v_k, v_l) | (v_i, v_j) \in \overline{E_X}, (v_k = v_l = v_Z) \vee \big((a(v_k) = a(v_i)) \wedge (e_{kl} \in C_X \vee e_{lk} \in C_X)\big)\}$.

(5) Integer variables $h_{ij} \in [0, H]$ for each tuple $(v_i, v_j) \in \overline{E_X}$, where $H = \max(|\mathcal{A}| - 2, |C_X|)$.

The constraints include the following, where $l_{ij} = -u_{ji}$:

(1) $\forall v_i, v_j \in V, u_{ij} + u_{ji} \geq 0$

(2) $\forall e_{ij} \in E_X, (l_{ij} \geq L_{ij}) \wedge (u_{ij} \leq U_{ij})$

(3) $\forall e_{ij} \in C_X, (0 \leq l_{ij} \leq L_{ij}) \wedge (u_{ij} \geq U_{ij})$

(4) $\forall e_{ij} \in E_X, (b_{ij} = 1) \wedge (b_{ji} = 1)$

(5) $\forall (v_i, v_j) \in \overline{E_X}, b_{ij} = \sum_{v_k, v_l | (v_i, v_j, v_k, v_l) \in Q} z_{ijkl}$

(6) $\forall (v_i, v_j, v_k, v_l) \in Q, u_{ij} \geq u_{ik} + u_{kl} + u_{lj} + (z_{ijkl} - 1)M,$ where M is a large constant

(7) $\forall (v_i, v_j, v_k, v_l) \in Q$ s.t. $(v_l, v_j) \in \overline{E_X}, z_{ijkl} \leq b_{lj}$

(8) $\forall (v_i, v_j, v_k, v_l) \in Q$ s.t. $(v_l, v_j) \in \overline{E_X}, h_{ij} + (1 - z_{ijkl})(H + 1) \geq h_{lj} + 1$

(9) $\forall e_{ij} \in C_X, \sum_{v_k | (v_i, v_j, v_k) \in T} c_{kj} = 1$

(10) $\forall (v_i, v_j, v_k) \in T, (u_{kj} \geq u_{ki} + u_{ij} + (c_{kj} - 1)M) \wedge (0 \leq l_{kj} \leq l_{ki} + l_{ij} + (1 - c_{kj})M)$

(11) $\forall (v_i, v_j, v_k) \in T$ s.t. $(v_i, v_k) \in \overline{E_X}, c_{kj} \leq b_{ik}$ and $\forall (v_i, v_j, v_k) \in T$ s.t. $(v_k, v_i) \in \overline{E_X}, c_{kj} \leq b_{ki}$

For our state temporal decoupling algorithm, we make an adaptation to the above encoding since we do not require all the communication links to be used. For example, in Figure 1, even though there is a communication link $ep_{EE'}$ from AUV1 to AUV2, it may not be used to support the decoupling of any external requirement episodes, in which case we do not need to satisfy any of its state constraints. Therefore, we additionally add a boolean variables p_j for each $ep_{ij} \in C_X$, which denotes whether the communication link is used or not. We add the following constraints:

- $\forall (v_i, v_j, v_k, v_l) \in Q$ s.t. $e_{kl} \in C_X, p_l \geq z_{ijkl}$ and $\forall (v_i, v_j, v_k, v_l) \in Q$ s.t. $e_{lk} \in C_X, p_k \geq z_{ijkl}$. This says that the communication link must be decoupled if it is used to support an external requirement constraint.

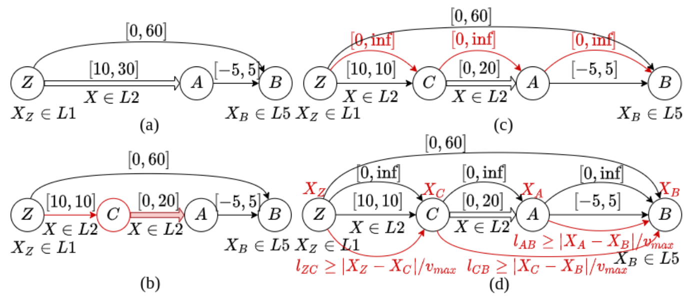

Figure 4: Feasibility for QSP with temporal certainty

- $\forall (v_i, v_j, v_k) \in T$ s.t. $e_{lk} \in C_X, p_k \geq c_{kj}$. This says that the communication link must be decoupled if it is used to support an external contingent constraint.

Additionally, if the end event of a communication link ep_{ij} has other constraints connected to it, then we need to set $p_j = 1$ as well. The constraint (9) in Casanova's encoding should be changed to $\forall e_{ij} \in C_X, \sum_{v_k | (v_i, v_j, v_k) \in T} c_{kj} = p_j$ so that only the used communication links are decoupled.

Feasibility Encoding for QSP

While the validity encoding ensure that the external episodes are decoupled, the imposed decoupling episodes may overconstrain the local state plans. For MaSTNU, Casanova et al. use the MILP dynamic controllability encoding proposed by Cui et al. to ensure the feasibility of local STNUs (Cui and Haslum 2017). We describe a novel feasibility checking algorithm for QSPs that builds on top of Cui's MILP encoding. While previous path planning algorithms exist for QSPs (Fernández-González, Williams, and Karpas 2018; Reeves, Fernández-González, and Williams 2019; Chen, Williams, and Fan 2021), they often assume a given total ordering of the events and that all the events are executable without any temporal uncertainty. As a result, their solution is typically a deterministic trajectory with a list of waypoints at fixed times. As mentioned, due to the existence of uncontrollable events, our execution strategy for a QSP is a dynamic policy.

To illustrate the high-level idea for finding an execution strategy for QSP, consider a simple QSP in Figure 4(a). First, we convert the QSP into its normal form, where every contingent temporal constraint has a lower bound of 0 (Morris 2006), and no contingent temporal constraint starts from an uncontrollable event, as shown in Figure 4(b). This can be achieved by rewriting each contingent episode into a requirement episode with a fixed duration equal to the original lower bound, followed by a contingent episode with a lower bound of 0. Second, we enforce a total ordering of the events at which state variables are constrained, as shown in Figure 4(c). Notice that we only need to order the events that have state constraints, since for any event without any state constraints, we do not care what values the state variables take at those events. Finally, given the ordering, we can express the reachability constraint by specifying how long it takes at least for the agent to go from one location to the next as the lower bound between those two events, as in Figure 4(d).

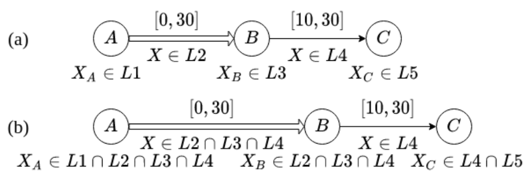

Figure 5: (a) Example QSP with 5 state constraints (b) Effective state constraints for example QSP

We then check the feasibility of QSP by checking its dynamic controllability. We now describe the MILP encoding in detail in the order of state constraints, ordering and reachability constraints, and dynamic controllability constraints.

(1) Encoding State and Delta Constraints First, we encode all the continuous state and delta constraints, which includes: $SC_{start}, SC_{end}, SC_{overall}, DC$. We denote the set of *constrained events* as $V_{sc} = \{v | v = s(ep(c)), \forall c \in SC_{start} \cup SC_{overall} \cup DC$ or $v = t(ep(c)), \forall c \in SC_{end} \cup SC_{overall} \cup DC\}$, where $ep(c)$ denotes the episode that the constraint c belongs to, and $s(ep)$, $t(ep)$ denote the start and end event of the episode ep, respectively. For each state variable $x \in X$ and for each constrained event $v_i \in V_{sc}$, we create a continuous variable x_i that represents the value of the state variable at that event. We denote X_i as the vector of continuous variables for state variables X at event v_i.

To encode the satisfaction of state constraints, consider the example in Figure 5(a) with five state constraints. In this case, since both $X \in L2$ and $X \in L4$ are overall state constraints to be satisfied throughout the episodes, their start and end events need to satisfy those state constraints too. Additionally, since ep_{AB} includes a contingent temporal constraint with a lower bound of 0, meaning that event v_B is an uncontrollable event that may occur any time on or after event v_A, any state constraint that needs to be satisfied at v_B must also be satisfied at v_A as well as throughout episode ep_{AB}. Therefore, the effective state constraints that need to be satisfied and encoded for V_{sc} are shown in Figure 5(b). Note that since we assume normal form of the QSP, event v_A cannot be another uncontrollable event and the propagation of state constraints due to contingent constraints is limited. Additionally, for any overall state constraint sc of an episode, any constrained event ordered in between the episode must also satisfy the constraint. Based on the MILP dynamic controllability encoding, l_{ij} denotes the continuous variable for the lower bound between event v_i and v_j. Therefore, we enforce the overall state constraint $sc \in SC_{overall}$ for an episode ep_{AB} using the following constraint:

(C1) $\forall v_D \in V_{sc}, (l_{DA} \geq 0) \vee (l_{BD} \geq 0) \vee sc(X_D)$.

Intuitively, this says that either v_D is ordered before v_A, or v_D is ordered after v_B, or the state constraint has to be satisfied at event v_D. Note that in the case of v_D being an uncontrollable event with contingent episode ep_{GD}, as mentioned, any state constraint that applies to v_D should apply as an overall state constraint for the entire episode ep_{GD}. If v_G is ordered before v_A, based on our reachability analysis in the following section, $X_A = X_G$ must hold to satisfy dynamic controllability, and any event v_E ordered in between ep_{GD} also satisfies $X_G = X_E$, which automatically satisfies the overall state constraint.

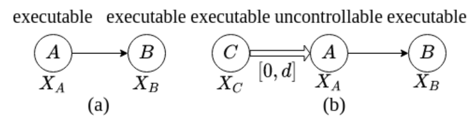

Figure 6: Examples for ordering and reachability constraints

(2) Encoding Ordering & Reachability In order to enforce a total ordering between constrained events and reachability between pairs of events, we can combine them into the following constraints. Note that we assume agents have first-order dynamics, and our total ordering is a weak total ordering that allows events to occur simultaneously.

(C2) $\forall v_A, v_B \in V_{sc}$ such that v_A, v_B are executable, $\forall x \in X, (l_{AB} \geq |x_A - x_B|/v_{max}) \vee (l_{BA} \geq |x_A - x_B|/v_{max})$

(C3) $\forall v_A, v_B \in V_{sc}$ such that v_A or v_B is uncontrollable, $(l_{AB} \geq 0) \vee (l_{BA} \geq 0)$

(C4) $\forall v_A, v_B \in V_{sc}$ such that v_A is uncontrollable and v_B is executable, $\forall x \in X, (l_{BA} \geq 0) \vee (l_{AB} \geq |x_A - x_B|/v_{max})$

For C2 (Figure 6(a)), when v_A and v_B are both executable events with state variable values X_A and X_B, then assuming first-order dynamics, we know it takes at least $\max_{x \in X} |x_A - x_B|/v_{max}$ time to go from one event to another. Therefore, we order them by imposing that the temporal lower bound either from v_A to v_B or from v_B to v_A has to be greater or equal to the above. For C3 (Figure 6(b)), when any of v_A or v_B is an uncontrollable event, then this constraint only enforces the ordering between the two events. C2 and C3 together enforces a global total ordering of all the constrained events. Note that any contingent episode such as ep_{CA} must satisfy $lb_{CA} \geq 0$ by definition, which satisfies C3. For C4 (Figure 6(b)), when event v_B is an executable event ordered after an uncontrollable event v_A, then there is a reachability constraint from v_A to v_B.

An example of enforced reachability constraints assuming given order can be seen in Figure 4(d). Note that we omitted the reachability constraint from v_Z to v_B to avoid cluttering, since it is dominated by other reachability constraints. Notice that without a reachaility constraint from v_C to v_A, X_A can take any value within the range of its state constraints, and we do not strictly require the agent to be at X_A at event v_A. To understand the execution strategy, we will focus on a minimal QSP in Figure 6(b) involving a contingent episode ep_{CA} and the first executable event v_B ordered after v_A, since the execution strategy for any consecutive executable events is simple. In order to describe the execution strategy, we will write the QSP's underlying temporal network in its labeled distance graph form (Morris 2006) in Figure 7, where l_{ij} (u_{ij}) denotes the lower bound (upper bound) from v_i to v_j. According to C2 and C4, we have reachability constraints $l_{CB} \geq |X_C - X_B|/v_{max}$ and $l_{AB} \geq |X_A - X_B|/v_{max}$. Assuming that X_A is constrained

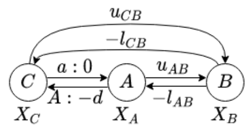

Figure 7: Execution policy in labeled distance graph

to be in region $L2$, then it means that X_C must be within region $L2$ too. The execution policy involves the agent starting at location X_C at event v_C, and going towards X_B. If the agent reaches the boundary for region $L2$, but hasn't received event v_A, then it will be stuck at the boundary until v_A occurs. When v_A occurs, the agent continues going towards X_B. The execution policy is feasible if the resulting network is dynamically controllable, because based on the dynamic controllability constraints:

- $u_{CB} - l_{CB} \geq 0$. This ensures that there is enough time on u_{CB} for the agent to go from X_C to X_B non-stop.
- $u_{CB} - l_{AB} - d \geq 0$. If the agent gets stuck at $L2$ boundary, then in the worst case, it needs to wait until d time has passed before continuing its way to X_B. This ensures that we can find such a location X_A within the $L2$ boundary such that there is enough time on u_{CB} for the agent to wait for d time and go from X_A to X_B.
- $u_{AB} - l_{AB} \geq 0$. This ensures that if the agent is stuck at $L2$ boundary until v_A occurs, there is enough time on u_{AB} for it to go from the boundary point X_A to X_B.
- $u_{AB} + 0 - l_{CB} \geq 0$. This ensures that if event v_A occurs immediately after event v_C, there is enough time on u_{AB} for the agent to go from X_C to X_B.

Note that it is possible for other events to be ordered in between v_C and v_A. If an executable event v_D is ordered in between v_C and v_A, then reachability and dynamic controllability constraints require that $X_C = X_D$. If an uncontrollable event v_E is ordered in between v_C and v_A, and its corresponding contingent episode is ep_{GE}, then it requires that $X_C = X_G$ and $v_C = v_E$, that is, event v_C is scheduled immediately when v_E is received. We leave it to the reader to validate the feasibility of execution policies in these cases.

(3) Encoding Dynamic Controllability We summarize the MILP constraints that ensure the dynamic controllability of the local plans (Cui and Haslum 2017; Wah and Xin 2007). Given a STNU $\langle V, E, C\rangle$, where V is the set of events, E is the set of temporal requirement constraints, and C is the set of temporal contingent constraints, the MILP formulation includes the following variables, where V_E denotes the set of executable events. Note that $l_{ij} = -u_{ji}$.

(1) Real variables u_{ij} for $v_i, v_j \in V$, with $u_{ii} = 0$

(2) Real variables w_{ijk} for $e_{ik} \in C, v_j \in V_E$, with $w_{iik} = 0$

The MILP constraints are listed below:

(1) $\forall e_{ij} \in E \cup C, (l_{ij} \geq L_{ij}) \wedge (u_{ij} \leq U_{ij})$

(2) $\forall e_{ij} \in C, (0 \leq l_{ij} \leq L_{ij}) \wedge (u_{ij} \geq U_{ij})$

(3) $\forall v_i, v_j, v_k \in V, u_{ij} \leq u_{ik} + u_{kj}$

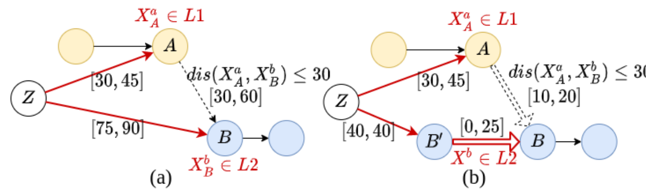

Figure 8: Decouple delta constraint for (a) external requirement episode (b) external contingent episode

(4) $\forall e_{ik} \in C, \forall v_j \in V_E, (l_{jk} < 0) \vee \big((u_{ij} \leq l_{ik} - l_{jk}) \wedge (l_{ij} \geq u_{ik} - u_{jk})\big)$

(5) $\forall e_{ik} \in C, \forall v_j \in V_E, u_{ik} - u_{jk} \leq w_{ijk}$

(6) $\forall e_{ik} \in C, \forall v_j \in V_E, \min(l_{ik}, w_{ijk}) \leq l_{ij}$

(7) $\forall e_{ik} \in C, \forall v_j, v_m \in V_E, w_{ijk} - u_{mj} \leq w_{imk}$

(8) $\forall e_{ik}, e_{mj} \in C, (w_{ijk} < 0) \vee (w_{ijk} - l_{mj} \leq w_{imk})$

When the external contingent episodes are decoupled, additional local contingent episodes may be introduced as part of the decoupling episodes. Therefore, the above encoding needs to be adapted to handle these optional local contingent constraints. Refer to (Casanova et al. 2016) for detail.

State Validity Encoding

To ensure the validity of decoupling, the external state and delta constraints across agents should be decoupled too. We describe how to decouple DC and $SC_{overall}$ next. For simplicity, we assume that the events of an external episode ep_{AB} are executable events, except for the uncontrollable end event when ep_{AB} is an external contingent episode. This assumption can be removed with some more analysis. When encoding state validity constraints, for a communication link ep_{ij}, we condition its MILP constraints on $p_j = 1$ so that they are only enforced when ep_{ij} is used.

First, as shown in Figure 8, consider an external episode ep_{AB} with a delta constraint $dc(X_A^a, X_B^b) \in DC$. To find the state decoupling for $dc(X_A^a, X_B^b)$, it suffices to find $L1$, $L2$ such that for any $X_A^a \in L1$ and for any $X_B^b \in L_2$, $dc(X_A^a, X_B^b)$ always holds. In this way, we have decoupled the external delta constraint by restricting locally for each agent an area that it should be in at the specific start and end event of the episode. Note that if ep_{AB} is a contingent episode (Figure 8(b)), then its decoupling involves an introduced local contingent episode ep_{ZB} in its normal form, or more specially, a requirement episode $ep_{ZB'}$ and a contingent episode $ep_{B'B}$. Because v_B is an uncontrollable event, $X^b \in L2$ has to be satisfied over the entire $ep_{B'B}$.

To encode the above constraint in MILP, while it is possible to directly find such regions $L1, L2$ approximated by a set of points as a polygon, in this paper, we directly enforce the constraint $dc(X_A^a, X_B^b)$. If ep_{AB} is a contingent episode, we additionally enforce $dc(X_A^a, X_{B'}^b)$. Once a MILP solution is found, we can read off region $L1$ as the point X_A^a, and region $L2$ as the point X_B^b or the region enclosed by X_B^b and $X_{B'}^b$ in the case of a contingent episode. As a post-processing step, we can optionally relax the region $L1$ and

$L2$ such that the decoupling is still valid to provide more flexibility to the agents.

Second, as shown in Figure 9(a), consider an external requirement episode ep_{AB} with an overall state constraint $sc(X^a, X^b) \in SC_{overall}$. In this case, the resulting decoupling introduces copies of event v_A and event v_B and external temporal constraints $e_{AA'}$ and $e_{B'B}$ with duration 0. The decoupling of $e_{AA'}$ and $e_{B'B}$ can be handled by the temporal validity encoding. We can similarly find regions $L1$ and $L2$ such that $\forall v_A \leq T_1 \leq v_{B'}, X^a_{T_1} \in L1$ for agent a and $\forall v_{A'} \leq T_2 \leq v_B, X^b_{T_2} \in L2$ for agent b, $sc(X^a_{T_1}, X^b_{T_2})$. In order to encode the above constraints in MILP, we enforce the constraints $sc(X^a_A, X^b_{A'})$, $sc(X^a_A, X^b_B)$, $sc(X^a_{B'}, X^b_{A'})$ and $sc(X^a_{B'}, X^b_B)$. Additionally, for any constrained events in between the two episodes, we enforce the overall state constraint: $\forall v^a_D \in V^a_{sc}, \forall v^b_E \in V^b_{sc}$, $(l_{DA} \geq 0) \vee (l_{B'D} \geq 0) \vee (l_{EA'} \geq 0) \vee (l_{BE} \geq 0) \vee sc(X^a_D, X^b_E)$. Finally, we can read off region $L1$ as the convex region enclosed by the state variable values in between v_A and $v_{B'}$, and similarly for $L2$. $L1, L2$ can also be relaxed in post-processing.

Finally, as shown in Figure 9(b), consider an external contingent episode ep_{AB} with an overall state constraint $sc(X^a, X^b) \in SC_{overall}$. The resulting decoupling involves a local requirement temporal constraint e_{ZA} and a local contingent episode ep_{AC} for agent a, where ep_{AC} has an overall state constraint $X^a \in L1$ that requires agent a to stay in region $L1$ throughout the communication period. Note that we assume in this case, agent a receives event v_C upon communication ep_{AB} finishes, since it is often used to model data transmission that takes up a period of time. The decoupling also involves a local requirement temporal constraint $e_{ZA'}$, a requirement episode $ep_{A'B'}$ and a contingent episode $ep_{B'B}$ for agent b, where $ep_{A'B'}$ and $ep_{B'B}$ are under the overall state constraint $X^b \in L2$. We encode the constraints in MILP in a similar fashion as before.

Preliminary Experiments

We evaluate our algorithm on two AUV team scenarios. All experiments were run on on 3.40GHZ 8-Core Intel Core i7-6700 CPU with 39GB RAM, and the MILP encoding was solved using Gurobi 9.1.2, with a timeout of 100 seconds. Note that the MILP encoding also allows the specification of an objective function, which affects the runtime. We evaluate the algorithm on three objective cases: (obj1) no objective function, (obj2) minimize the use of communication links, (obj3) minimize $\sum_{v_i \in V} u_{Zi}$.

For our motivating example, we record the average runtime for 3 communication scenarios: (ST1) only $ep_{EE'}$ is available, which the algorithm finds no decoupling solution, (ST2) $ep_{FF'}$ and $ep_{GG'}$ are available, and (ST3) all communication links are available. We also test the example for temporal decoupling only by framing it as a MaSTNU (T3), and for feasibility only by framing it as a QSP (QSP) that assumes full observability between agents. We repeat the experiments by adding two other missions to each AUV (tests denoted by *). The result is shown in Table 1. The results show that the choice of objective functions can have a large impact on the runtime. Additionally, the runtime increases quite drastically as the number of communication links increases. Note that for temporal decoupling, there exists a solution even with only $ep_{EE'}$, and the added missions are not totally ordered since no state constraints are enforced.

test	obj1	obj2	obj3	test	obj1	obj2	obj3
ST1	0.24	0.21	0.17	ST1*	2.87	3.33	5.28
ST2	0.4	0.3	0.71	ST2*	4.46	3.15	43.14(1)
ST3	4.23	2.56	24.9	ST3*	18.6(1)	47.39(4)	N/A(10)
T3	0.09	0.06	0.94	T3*	0.19	0.19	2.31
QSP	0.04	N/A	0.04	QSP*	6.84	N/A	23.6

Table 1: Average runtime in seconds over 10 runs for different tests, with the number of timed out runs in parenthesis

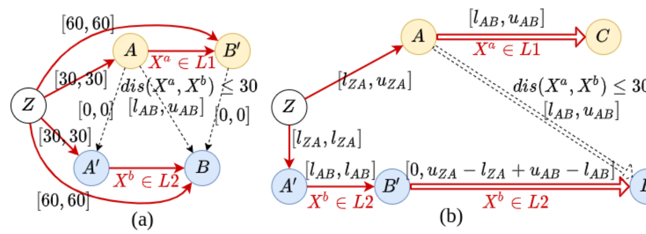

Figure 9: Decouple state constraint of type $SC_{overall}$ for (a) external requirement episode (b) external contingent episode

In a second scenario, the ship deploys an AUV in a region, and the AUV carries out two sampling missions at different science sites. When both missions are done, the AUV transmits data through a communication link back to the ship. The transmission may take any time between 20 to 30 minutes, during which they have to stay within 30 meters of each other. The AUV and the ship needs to stay within 100 meters of each other throughout the two sampling missions, and the ship has its own imaging mission to do before a certain deadline such that it has to be carried out concurrently while the AUV is on its sampling missions. Finding a solution takes 1.1 secs on average with obj1 and 1.6 secs with obj3.

Note that our QSP feasibility encoding finds an execution strategy that fixes the state variable values at constrained executable events, meaning for an executable event $v_i \in V_{sc}$ following an uncontrollable event $v_j \in V_{sc}$ with $[0, 0]$ temporal bound, we exclude any execution strategies where state variables at v_i can take any non-deterministic value that is taken at v_j. Future work can investigate if this assumption can be relaxed. we also assume no obstacles in the environment and simple dynamics of the vehicles. Future work can build on top of our encoding to allow richer path planning constraints and agent dynamics.

Conclusion

In this paper, we introduced the framework of Multi-Agent Qualitative State Plan (MaQSP) to model multi-agent plans with coupled temporal and state constraints, where agents are subject to limited communication during execution. We proposed a state temporal decoupling algorithm for MaQSP based on MILP encoding, which includes a novel path planning algorithm for QSPs with temporal uncertainty that may be useful in other applications.

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0035. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

References

Boerkoel Jr, J. C.; and Durfee, E. H. 2013. Distributed reasoning for multiagent simple temporal problems. *Journal of Artificial Intelligence Research* 47: 95–156.

Casanova, G.; Pralet, C.; Lesire, C.; and Vidal, T. 2016. Solving dynamic controllability problem of multi-agent plans with uncertainty using mixed integer linear programming. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, 930–938. IOS Press.

Chen, J.; Williams, B. C.; and Fan, C. 2021. Optimal Mixed Discrete-Continuous Planning for Linear Hybrid Systems. In *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, HSCC '21. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383394. doi:10.1145/3447928.3456654. URL https://doi.org/10.1145/3447928.3456654.

Cui, J.; and Haslum, P. 2017. Dynamic controllability of controllable conditional temporal problems with uncertainty. In *27th International Conference on Automated Planning and Scheduling (ICAPS 2017)*.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial intelligence* 49(1-3): 61–95.

Fernández-González, E.; Williams, B.; and Karpas, E. 2018. ScottyActivity: Mixed Discrete-Continuous Planning with Convex Optimization. *J. Artif. Intell. Res.* 62: 579–664.

Hunsberger, L. 2002. Algorithms for a temporal decoupling problem in multi-agent planning. In *AAAI/IAAI*.

Léauté, T.; and Williams, B. C. 2005. Coordinating agile systems through the model-based execution of temporal plans. In *AAAI*, 114–120.

Morris, P. 2006. A structural characterization of temporal dynamic controllability. In *International Conference on Principles and Practice of Constraint Programming*, 375–389. Springer.

Reeves, M.; Fernández-González, E.; and Williams, B. 2019. Executing Multi-Goal Mission Plans for Coordinated Mobile Robots. In *ICAPS 2019 INTEX workshop*.

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

Wah, B. W.; and Xin, D. 2007. Optimization of bounds in temporal flexible plans with dynamic controllability. *International Journal on Artificial Intelligence Tools* 16(01): 17–44.

Zhang, Y.; and Williams, B. C. 2021. Privacy-Preserving Algorithm for Decoupling of Multi-Agent Plans with Uncertainty. In *Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS)*.