

Dynamic Quantitative Skill Distribution in Multi-Agent Temporal Planning as SMT

Josselin Guéron 

Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ, GREYC UMR6072,
F-14000 Caen, France

Frédéric Maris 

Université de Toulouse, IRIT, France

Abstract

We introduce a new framework for multi-agent temporal planning, based on event-based temporal planning, dynamic team formation and Multi-Robot Task Allocation (MRTA). Unlike existing approaches, team formation is directly integrated into the planning and scheduling process. Then, both the teams and agents' skills may evolve over time. In particular, our framework allows us to address complex and dynamic multi-agent problems, such as cooperative task execution, where heterogeneous agents with qualitative and quantitative skills must cooperate dynamically. We propose an SMT encoding of these team temporal planning problems that utilizes the linear arithmetic theory over the rational numbers, QF-LRA. Next, to demonstrate that our approach is suitable for real-world applications, we present experimental results on disaster response problems.

2012 ACM Subject Classification Computing methodologies → Planning and scheduling

Keywords and phrases Multi-Agent Temporal Planning, Multi-Agent Task Allocation, SMT-based Planning, Dynamic Team Formation

Digital Object Identifier 10.4230/OASICS.TIME.2026.9

1 Introduction

The deployment of autonomous agents in complex, real-world environments, such as search-and-rescue operations, automated manufacturing, and space exploration, relies increasingly on Multi-Agent Systems (MAS). In these domains, tasks are often too complex, heavy, or specialized to be executed by a single agent. Consequently, heterogeneous agents must cooperate by forming temporary groups to fulfill specific mission requirements. This challenge falls at the intersection of Multi-Agent Task Allocation (MRTA), dynamic team formation and temporal planning.

A critical aspect of modeling heterogeneous agents lies in accurately capturing their capabilities. In this paper, we distinguish between two fundamental types of skills. Qualitative skills represent static, innate capabilities (e.g. possessing a specific sensor) that an agent either masters or does not. Conversely, quantitative skills represent measurable, bounded capacities (e.g. lifting capacity or battery level) that can be aggregated among agents and consumed or produced over time.

Furthermore, real-world tasks are rarely instantaneous. They are dynamic processes that unfold over time, requiring different skills at different stages and inducing specific effects at precise moments. While existing literature on task-driven team formation extensively covers the assignment of agents to tasks based on skill matching, it often abstracts away the fine-grained temporal constraints of task execution. On the other hand, expressive temporal planning frameworks handle complex temporal and numeric constraints but traditionally struggle with the combinatorial explosion of dynamically forming teams. Encoding such heavily numeric and temporal problems into classical boolean Satisfiability (SAT) is notoriously inefficient due to the discretization of time and resources.



© Josselin Guéron and Frédéric Maris;

licensed under Creative Commons License CC-BY 4.0

33rd International Symposium on Temporal Representation and Reasoning (TIME 2026).

Editors: AndreA Orlandini and Sophie Pinchinat; Article No. 9; pp. 9:1–9:20

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

To bridge this gap, this paper introduces a novel framework for Multi-Agent Temporal Planning Problems. We propose an event-based temporal task model that seamlessly integrates dynamic team formation with fine-grained temporal and numeric reasoning. By leveraging Satisfiability Modulo Theories (SMT), we map the discrete logic of qualitative skill alignment to the boolean engine, while delegating the continuous temporal constraints and quantitative resource aggregations to specialized arithmetic theory solvers.

Specifically, the main contributions of this paper are the following:

- We formalize a temporal multi-agent planning domain encompassing heterogeneous agent classes with qualitative and quantitative skills.
- We extend classical event-based temporal semantics to support task-driven team formation, defining strict applicability conditions based on skill requirements and temporal effects.
- We present an efficient SMT encoding that natively resolves the complex interaction between discrete team assignments and the numeric constraints of temporal execution, validated by experimental results on disaster response problems.

This paper is organized as follows: Section 2 covers related work across SMT, temporal planning, MRTA, and dynamic team formation. Section 3 formalizes the multi-agent temporal planning problem, while Section 4 describes its SMT encoding. Section 5 evaluates the framework through experiments in a disaster response domain. Finally, Section 6 concludes this work and outlines future research directions.

2 Related Work

2.1 Satisfiability Modulo Theory (SMT)

Satisfiability Modulo Theories (SMT) extends the classic boolean Satisfiability (SAT) problem by interpreting symbols with respect to background theories, such as Linear Integer Arithmetic (LIA) or Linear Real Arithmetic (LRA) [1]. Traditional SAT solvers strictly operate on propositional formulas, while SMT solvers can natively evaluate logical formulas containing complex numeric and arithmetic constraints (e.g. $x + y \leq z$). Although SAT is widely used in classical automated planning, SAT-based approaches inherently struggle with two major aspects of real-world dynamic domains: temporal flexibility (continuous or dense time) and numeric resources [14]. This limitation arises from the severe combinatorial explosion caused by discretizing large numeric domains, quantitative capacities, or real-valued timepoints into purely boolean variables.

SMT overcomes these limitations by delegating arithmetic reasoning to specialized theory solvers, allowing for scalable reasoning over continuous time and quantitative state variables. In the context of temporal and numeric planning, SMT has been successfully used to encode highly expressive formalisms such as PDDL2.1. Approaches like SMTPlan+ [4] and works in temporal planning [13] demonstrate that SMT is particularly well-suited for event-based planning. In these encodings, real-valued or integer variables represent the exact occurrence times of events (e.g. τ_i) and the levels of numeric resources. The temporal relationships between tasks are efficiently encoded using Difference Logic, while resource consumptions and productions are modeled using Linear Arithmetic.

Our multi-agent temporal planning problem aligns perfectly with the capabilities of SMT. The event-based framework introduced in the previous sections relies simultaneously on strict logical prerequisites and cumulative numeric constraints. Encoding this model into SMT allows us to jointly and efficiently solve two interdependent problems:

- The discrete logic of task assignments and team formation based on qualitative skill matching, natively handled by the boolean SAT engine.
- The arithmetic aggregation of quantitative skills (e.g. ensuring team resources meet task requirements) and the resolution of the relative temporal constraints over event times, natively handled by the LRA theory solver.

2.2 Event-based Temporal Planning

To model durative actions and continuous time, we rely on an event-based temporal planning framework similar to that given in [6]. Rather than treating actions as atomic units with fixed durations, this framework decomposes each action into a set of instantaneous *events* (e.g. starting an action, establishing a fact, or ending a condition) scheduled over a temporal horizon.

State variables representing properties of the environment that evolve over time are modeled as boolean *fluents*. An action (or action-instance) a is defined by its constituent events, denoted $Events(a)$, which interact with fluents in two ways:

- Effects: The event $a \rightarrow f$ establishes fluent f (making it true), while $a \rightarrow \neg f$ destroys f (making it false).
- Conditions: The notations $f \mid \rightarrow a$ and $f \rightarrow \mid a$ represent the exact start and end timepoints, respectively, of the continuous interval over which fluent f must remain true for action a to be valid.

A real-valued function $\tau(E) \in \mathbb{R}$ assigns the absolute execution time to any event E . The internal temporal structure of an action a is governed by relative temporal constraints $Constr(a)$. For any pair of events $e_1, e_2 \in Events(a)$, their relative time difference is bounded by an interval $[\alpha_a(e_1, e_2), \beta_a(e_1, e_2)]$, imposing that $\tau(e_1) - \tau(e_2) \in [\alpha_a(e_1, e_2), \beta_a(e_1, e_2)]$. This mechanism naturally models variable action durations and internal event orderings.

A valid temporal plan must satisfy two fundamental categories of constraints:

1. Inherent Constraints: Every action instance $a \in A$ must respect its internal timing rules defined by $Constr(a)$.
2. Contradictory-Effects Constraints: To prevent physical inconsistencies, no two events occurring at the exact same instant can establish and destroy the same fluent. Formally, for any pair of action instances $a_i, a_j \in A$ and any fluent $f \in Del(a_i) \cap Add(a_j)$, we require that $\tau(a_i \rightarrow \neg f) \neq \tau(a_j \rightarrow f)$.

► **Definition 1** (Temporal Planning Problem). A temporal planning problem is a tuple $\langle I, A, G \rangle$ where I is the initial set of true fluents, G is the set of goal fluents, and A is the set of actions.

► **Definition 2** (Temporal Plan). A temporal plan for a problem $\langle I, A, G \rangle$ is a pair $P = \langle A, \tau \rangle$, where A is a finite set of action-instances $\{a_1, \dots, a_n\}$ and τ is a real-valued function on $Events(A)$, is a temporal plan for the problem $\langle I, A', G \rangle$ if (1) $A \subseteq A'$, and (2) P satisfies the inherent and contradictory-effect constraints on A ; and when P is executed (i.e. fluents are established or destroyed at the times given by τ) starting from the initial state I : (3) for all $a_i \in A$, each $f \in Cond(a_i)$ is true when it is required, and (4) all goal fluents $g \in G$ are true at the end of the execution of P .

Events are instantaneous, whereas actions are not only durative but may also be of variable length. Thus a temporal plan P does not schedule its action-instances directly but schedules all the events in its action-instances.

2.3 Multi-Robot Task Allocation

Multi-Robot Task Allocation (MRTA) is a coordination problem where a group of agents must efficiently assign themselves to tasks in order to perform them.

► **Definition 3** (MRTA framework). *Formally, the MRTA framework is defined in the literature as a tuple $\langle N, T, u \rangle$, such that:*

- $N = \{r_1, r_2, \dots, r_n\}$, a set of n robots.
- $T = \{t_1, t_2, \dots, t_m\}$, a set of m tasks.
- $u : N \times T \rightarrow \mathbb{R}$, a utility function that assigns the value of performing a task to a robot. The resulting value u_{ij} represents the utility gained (or cost incurred, as negative utility) from robot r_i completing task t_j . In the extended case where tasks require multiple robots, this utility function is typically generalized over subsets of agents: $u : 2^N \times T \rightarrow \mathbb{R}$.

The more standard formalism of MRTA defines this problem in a “one-to-one” way: one agent can be assigned to one task, and one task can be assigned by one agent. However, as defined in [9], there is a comprehensive taxonomy allowing us to describe more complex types of problems. This taxonomy is defined following three axes:

- Single-task (ST) vs Multi-task (MT) robots: ST means that one robot can only perform one task simultaneously, while MT means that one robot can perform one or more task simultaneously.
- Single-robot (SR) vs Multi-robot (MR) tasks: SR means that a task can be performed by a single agent, while MR means that multiple robots can be required to perform the task.
- Instantaneous (IA) vs Time-extended (TA) assignment: IA means that we can only create an assignment for the next execution phase, without future planning. TA means that we can schedule assignments in advance, before sequential and/or asynchronous executions.

According to this taxonomy, the challenge addressed in this paper, i.e. of coordinating a fleet of heterogeneous agents to execute complex tasks through an automated planning approach, falls into the ST-MR-TA category.

2.4 Dynamic Team Formation

While classic MRTA algorithms excel at optimizing utility and task scheduling, they often treat the required groups as black boxes. To address the inner mechanics of group composition, the Multi-Agent Systems (MAS) literature proposed the field of *Dynamic Team Formation* [16, 8]. Dynamic team formation focuses on cooperative, short-lived and task-driven alliances between agents. This framework is highly relevant for critical and high-uncertainty domains such as Search and Rescue (SAR), where teams of heterogeneous robots must be formed on-the-fly to combine complementary capabilities against emerging disaster events [11]. In these scenarios, teams dissolve immediately upon task completion so that individual assets can be fluidly redistributed to other pressing sub-goals [8].

To ensure the feasibility of these dynamic teams, the allocation framework must explicitly model the capabilities of heterogeneous agents. In the literature, this is achieved by matching agent profiles against task requirements using two distinct paradigms:

- Qualitative Skill Alignment: Tasks specify a set of mandatory operational roles or technical skills (e.g. search capability, specific sensor). A formed team is valid if and only if the joint capabilities of its assigned members satisfy the qualitative prerequisites of the task, ensuring that all required roles are covered [5].

- **Quantitative Resource Aggregation:** Beyond binary skills or roles, agents possess bounded, numeric resources (e.g. payload capacity, battery power). The quantitative capability of a team is modeled as the cumulative sum of the individual resources contributed by its members, which must satisfy or exceed the task's dynamic requirement threshold [15].

Despite the maturity of both MRTA and dynamic team formation, a significant gap remains at their intersection with formal temporal planning. Most existing team formation algorithms operate under large-scale temporal abstractions, treating tasks as blocks with static durations and fixed resource costs. Conversely, expressive temporal planners can schedule complex timelines but struggle with the combinatorial explosion induced by the dynamism of heterogeneous teams. Our work bridges this gap by unifying dynamic, skill-driven team formation with an event-based temporal planning framework, leveraging the efficiency of SMT solvers to handle the underlying numeric and logical constraints.

3 Multi-Agent Temporal Planning

We address a Multi-Agent Temporal Planning Problem. Briefly, this involves multi-agent planning within a dynamic and temporal environment, where the cooperation between heterogeneous agents is required to successfully execute a plan.

3.1 Agents and Mastered Skills over Time

► **Definition 4 (Environment).** *Let us define the multi-agent environment of our problem as a tuple $\mathcal{E} = \langle N, \mathbb{T} \rangle$, where $N = \{n_1, \dots, n_m\}$ is a set of m agents and $\mathbb{T} \in \{\mathbb{Z}, \mathbb{Q}\}$ represents the temporal domain of the environment (discrete or continuous).*

Agents and tasks are characterized by a set of skills K , partitioned into qualitative and quantitative types based on their temporal evolution and aggregation properties.

► **Definition 5 (Skills).** *Let $K = K_{ql} \cup K_{qt}$ be the set of all available skills, defined as the disjoint union of qualitative (boolean) skills $K_{ql} = \{k_1, \dots, k_l\}$ and quantitative (measurable) skills $K_{qt} = \{k_{l+1}, \dots, k_{l+l'}\}$.*

► **Example 6 (Running domain).** Throughout this section, we illustrate the definitions on a simplified Disaster Response domain with $K_{ql} = \{\text{Fire extinguishing}\}$ and $K_{qt} = \{\text{Water Capacity}\}$.

To model agent heterogeneity, we group agents into stationary *classes* (e.g. operational roles) with fixed profiles of mastered skills that cannot be acquired or lost over time.

► **Definition 7 (Agent Class).** *An agent class $w \in \mathcal{W}$ is defined as a tuple $w = \langle N^w, K_{ql}^w, K_{qt}^w, P_{qt}^w \rangle$ where:*

- $N^w \subseteq N$ is a subset of agents that belongs to the class w ,
- $K_{ql}^w \subseteq K_{ql}$ is a subset of qualitative skills that agents in this class have mastered,
- $K_{qt}^w \subseteq K_{qt}$ is a subset of quantitative skills that agents in this class have mastered,
- $P_{qt}^w : K_{qt}^w \rightarrow \mathbb{Q}_+$ is a capacity function mapping each quantitative skill $k \in K_{qt}^w$ to its maximum capacity for an agent of class w .

► **Example 8 (Agent Profile).** In a Disaster Response scenario, an agent class $w = \text{Firefighting Drone}$ could be defined with $K_{ql}^w = \{\text{Fire extinguishing}\}$, and a bounded quantitative skill $K_{qt}^w = \{\text{Water Capacity}\}$ with $P_{qt}^w(\text{Water Capacity}) = 100L$.

Qualitative skills represent static capabilities required to participate in a task. They are evaluated via a class-dependent indicator function.

► **Definition 9** (Qualitative skills function). *Let $m^{01} : N \times K_{qt} \rightarrow \{0,1\}$ be the indicator function determining whether an agent $n \in N^w$ of class $w \in \mathcal{W}$ masters a skill $k \in K_{qt}$:*

$$m^{01}(n, k) = \begin{cases} 1 & \text{if } k \in K_{qt}^w \\ 0 & \text{otherwise} \end{cases}$$

► **Example 10** (Qualitative instantiation). For n a *Firefighting Drone*, $m^{01}(n, \text{Fire extinguishing}) = 1$, while $m^{01}(n, k) = 0$ for any other qualitative skill $k \in K_{qt}$.

Accordingly, $m_n^{01} = \{k \in K_{qt} \mid m^{01}(n, k) = 1\}$ denotes the skills of agent n , and $m_C^{01} = \bigcup_{n \in C} m_n^{01}$ defines the qualitative skills aggregated by a team of agents $C \subseteq N$ via set union. Conversely, quantitative skills describe bounded capacities (e.g. resource) whose operational values can dynamically evolve but are bounded by the agent's class capacity.

► **Definition 11** (Quantitative skills function). *Let $m^Q : N \times K_{qt} \times \mathbb{T} \rightarrow \mathbb{Q}_+$ be the dynamic function determining the level of a quantitative skill for agent $n \in N_w$ of class $w \in \mathcal{W}$ at time $\tau \in \mathbb{T}$:*

$$\forall k \in K_{qt}, \quad 0 \leq m^Q(n, k, \tau) \leq \begin{cases} P_{qt}^w(k) & \text{if } k \in K_{qt}^w \\ 0 & \text{otherwise} \end{cases}$$

► **Example 12** (Quantitative instantiation). At time τ , a *Firefighting Drone* n with a half-full tank has $m^Q(n, \text{Water Capacity}, \tau) = 50 \leq P_{qt}^w(\text{Water Capacity}) = 100$.

The current profile of agent n is denoted $m_n^Q(\tau) = \{(k, m^Q(n, k, \tau)) \mid k \in K_{qt}^w \wedge m^Q(n, k, \tau) > 0\}$, and the aggregated quantitative skills of a team $C \subseteq N$ are modeled via arithmetic sum: $m_C^Q(\tau) = \{(k, v) \mid k \in K_{qt}, v = \sum_{n \in C} m^Q(n, k, \tau)\}$.

3.2 Event-Based Tasks

We extend the event-based temporal planning framework in [6], representing plans as networks of primitive events occurring at specific timepoints. We use a placeholder dot ($\cdot$) to denote any entity (e.g. a task t or Nature) interacting with the environment state.

► **Definition 13** (Primitive Events). *Let $f \in \mathcal{F}$ be a fluent, and $k_{qt} \in K_{qt}$, and $k_{qt} \in K_{qt}$ with a value $v \in \mathbb{Q}_+$ respectively a qualitative and a quantitative skill. We define the following primitive events:*

- *Fluent events:* $\cdot \rightarrow f$ (establish f), $\cdot \rightarrow \neg f$ (destroy f), and $f \rightarrow \cdot$ (require f).
- *Skill events:* $k_{qt} \rightarrow \cdot$ (require k_{qt}), $(k_{qt} \geq v) \rightarrow \cdot$ (require capacity v), $\cdot \rightarrow k_{qt}(+v)$ (produce capacity v), and $\cdot \rightarrow k_{qt}(-v)$ (consume capacity v).

► **Example 14** (Primitive events). The task $t = \text{Extinguish Fire}$ requires the qualitative skill via $\text{Fire extinguishing} \rightarrow t$, requires an aggregated capacity of 150L via $(\text{Water Capacity} \geq 150) \rightarrow t$, and later consumes it via $t \rightarrow \text{Water Capacity}(-150)$.

When a specific task $t \in T$ triggers an event, the placeholder $\cdot$ is replaced by t . Tasks are modeled as collections of temporal requirements and effects over an inherent timeline.

► **Definition 15** (Task Events). *For a task $t \in T$ and a relative timepoint $\tau \in \mathbb{T}$ at which the event occurs, we define the following profile functions:*

- *Fluent requirement:* $\text{cond}_f : T \times \mathcal{F} \times \mathbb{T} \rightarrow \{0,1\}$ indicates if $f \in \mathcal{F}$ is necessary (1) or not (0).

- *Qualitative requirement:* $r^{01} : T \times K_{ql} \times \mathbb{T} \rightarrow \{0, 1\}$ indicates if $k \in K_{ql}$ is necessary (1) or not (0).
- *Quantitative requirement:* $r^Q : T \times K_{qt} \times \mathbb{T} \rightarrow \mathbb{Q}_+$ specifies the minimum capacity needed for $k \in K_{qt}$.
- *Fluent effect:* $\text{eff}_f : T \times \mathcal{F} \times \mathbb{T} \rightarrow \{0, 1\}$ indicates whether fluent f is established (1) or destroyed (0).
- *Quantitative production:* $\text{add}_{qt} : T \times K_{qt} \times \mathbb{T} \rightarrow \mathbb{Q}_+$ represents the capacity produced.
- *Quantitative consumption:* $\text{del}_{qt} : T \times K_{qt} \times \mathbb{T} \rightarrow \mathbb{Q}_+$ represents the capacity consumed.

► **Example 16** (Task profile instantiation). For $t = \text{Extinguish Fire}$: at $\tau = 0$, $r^{01}(t, \text{Fire extinguishing}, 0) = 1$ and $r^Q(t, \text{Water Capacity}, 0) = 150$; at $\tau = 2$, $\text{del}_{qt}(t, \text{Water Capacity}, 2) = 150$ (the capacity is consumed once the fire is extinguished).

We map the occurrence times of these structural conditions and effects directly to the execution plan. Let $\tau_i(e)$ denote the relative timepoint of the i -th instance of event e (simplified to $\tau(e)$ if it occurs exactly once).

► **Definition 17** (Event Mapping). Let $t \in T$ be a temporal task. The relationship between the syntactic events of t and its temporal profile functions is defined as follows:

$$\begin{aligned}
 \tau(f \rightarrow t) = \tau_i &\iff \text{cond}_f(t, f, \tau_i) = 1 \\
 \tau(k_{ql} \rightarrow t) = \tau_i &\iff r^{01}(t, k_{ql}, \tau_i) = 1 \\
 \tau((k_{qt} \geq v) \rightarrow t) = \tau_i &\iff r^Q(t, k_{qt}, \tau_i) \geq v \\
 \tau(t \rightarrow f) = \tau_i &\iff \text{eff}_f(t, f, \tau_i) = 1 \\
 \tau(t \rightarrow \neg f) = \tau_i &\iff \text{eff}_f(t, f, \tau_i) = 0 \\
 \tau(t \rightarrow k_{qt}(+v)) = \tau_i &\iff \text{add}_{qt}(t, k_{qt}, \tau_i) = v \\
 \tau(t \rightarrow k_{qt}(-v)) = \tau_i &\iff \text{del}_{qt}(t, k_{qt}, \tau_i) = v
 \end{aligned}$$

Internal temporal constraints Constr_t are expressed as relations over these event times (e.g. $\tau((k_{qt} \geq v) \rightarrow t) \leq \tau(t \rightarrow k_{qt}(-v))$ ensures that capacity verification precedes consumption). Tasks can only be executed if the assigned team jointly satisfies these prerequisites.

► **Definition 18** (Executable Task Condition). A task $t \in T$ is executable by a team $C \subseteq N$ at time $\tau \in \mathbb{T}$ if and only if it satisfies both:

- *Qualitative validity:* $\forall k \in K_{ql}, r^{01}(t, k, \tau) = 1 \implies k \in m_C^{01}$
- *Quantitative validity:* $\forall k \in K_{qt}, r^Q(t, k, \tau) \leq \sum_{n \in C} m^Q(n, k, \tau)$

We denote $T_C(\tau) \subseteq T$ as the subset of tasks that are executable by the team C at time τ .

► **Example 19** (Task Execution). Consider a task $t = \text{Extinguish Fire}$. It requires the qualitative skill $r^{01}(t, \text{Fire Extinguishing}, \tau) = 1$ to be able spread water, and a quantitative aggregate capacity $r^Q(t, \text{Water Capacity}, \tau) \geq 150L$. A single *Firefighting Drone* (max 100L) cannot execute this task alone. The Executable Task Condition forces the dynamic formation of a team C containing at least two drones to satisfy the 150L threshold, alongside an agent providing the *Fire Extinguishing* capability.

► **Definition 20** (Temporal State). The dynamic state of the environment at time $\tau \in \mathbb{T}$ is a tuple $S(\tau) = \langle \mathcal{C}(\tau), m^Q(\tau), F(\tau) \rangle$, where $\mathcal{C}(\tau)$ is the active team structure (a partition of N), $m^Q(\tau)$ tracks individual quantitative skill levels, and $F(\tau) \subseteq \mathcal{F}$ is the set of true fluents.

3.3 Temporal Task Model and Team Formation Process

To formalize the problem, we first define the overall model, which encapsulates agents, skills, and tasks.

► **Definition 21** (Temporal Task Model). A Temporal Task Model (*TTM*) is a tuple $\mathcal{M} = \langle \mathcal{W}, N, T, K, \mathcal{F}, \mathbb{T} \rangle$ where $\mathcal{W}$ is the set of agent classes, N is the set of m agents partitioned into $\mathcal{W}$, T is the set of temporal tasks, $K = K_{ql} \cup K_{qt}$ is the partitioned set of skills, $\mathcal{F}$ is the set of propositional fluents, and $\mathbb{T}$ is the temporal domain.

As previously mentioned, in our environment, a task is not instantaneous. Its conditions and effects are tied to relative timepoints τ representing the progression of the task.

► **Definition 22** (Temporal Task). A temporal task $t \in T$ is a tuple $t = \langle E_t, \text{Constr}_t \rangle$, where $E_t = \langle \text{Cond}_t, \text{Eff}_t \rangle$ encapsulates the task events, and Constr_t represents the internal synchronization and duration constraints over its relative timepoints τ :

- Task Conditions (Cond_t):
 - $\text{cond}_f(t, f, \tau) \in \{0, 1\}$ indicates whether fluent $f \in \mathcal{F}$ must be true (1) at timepoint τ .
 - $r^{01}(t, k, \tau) \in \{0, 1\}$ indicates if qualitative skill $k \in K_{ql}$ is required.
 - $r^Q(t, k, \tau) \in \mathbb{Q}_+$ specifies the minimum required aggregate level of quantitative skill $k \in K_{qt}$.
- Task Effects (Eff_t):
 - $\text{eff}_f(t, f, \tau) \in \{0, 1\}$ dictates whether fluent f is established (1) or destroyed (0) at timepoint τ .
 - $\text{add}_{qt}(t, k, \tau) \in \mathbb{Q}_+$ and $\text{del}_{qt}(t, k, \tau) \in \mathbb{Q}_+$ respectively define the quantitative skill amounts produced or consumed at timepoint τ .

► **Definition 23** (Team Formation Process). Let $\mathcal{M}$ be a *TTM*. A Team Formation Process (*TFP*) is a tuple $p = \langle C, X, \mathcal{T}_b, \mathcal{T}_d, \mathcal{T}_e \rangle$ where:

- $C \subseteq N$ is the subset of agents allocated to the process.
- $X \subseteq T$ is the subset of tasks to be executed by the team.
- $\mathcal{T}_b \leq \mathcal{T}_d \leq \mathcal{T}_e$ are three ordered global timepoints representing the assembly phase ($\mathcal{T}_b$), the effective deployment ($\mathcal{T}_d$), and the dissolution of the team ($\mathcal{T}_e$).

For any task $t \in X$, $\mathcal{T}_b^t$ and $\mathcal{T}_e^t$ denote its specific start and end execution times, satisfying $\mathcal{T}_d \leq \mathcal{T}_b^t \leq \mathcal{T}_e^t \leq \mathcal{T}_e$.

► **Example 24** (Team Formation Process). Continuing the running example, a TFP $p = \langle \{n_1, n_2\}, \{\text{Extinguish Fire}\}, 0, 1, 3 \rangle$ assembles two *Firefighting Drones* n_1, n_2 at $\mathcal{T}_b = 0$, deploys them as a unified team from $\mathcal{T}_d = 1$, during which they jointly satisfy the 150L requirement of *Extinguish Fire* ($100L + 100L \geq 150L$), and dissolves the team at $\mathcal{T}_e = 3$.

A TFP is applicable in a state $S(\mathcal{T}_b)$ if all participating agents are initially free, i.e. they belong to singleton sets in the active team structure: $\forall n_i \in C, \{n_i\} \in \mathcal{C}(\mathcal{T}_b)$. Once triggered, the process enforces the following structural state transitions:

- $\forall n_i \in C, \{n_i\} \in \mathcal{C}(\tau)$ is maintained for all states during the assembly interval $\tau \in [\mathcal{T}_b, \mathcal{T}_d]$;
- $C \in \mathcal{C}(\tau)$ represents the unified operational team for all states during execution $\tau \in [\mathcal{T}_d, \mathcal{T}_e]$;
- $\forall n_i \in C, \{n_i\} \in \mathcal{C}(\mathcal{T}_e)$ models the instant dissolution of the team back into individual assets.

3.4 Multi-Agent Temporal Planning Problem

► **Definition 25** (Team Temporal Planning Problem). A Team Temporal Planning Problem (TTPP) is a tuple $\langle \mathcal{M}, I, G \rangle$ where $\mathcal{M}$ is a Temporal Task Model (TTM), $I = \langle C_0, m_0^Q, F_0 \rangle$ is the initial temporal state, and $G \subseteq \mathcal{F}$ the set of goal fluents.

To ensure physical feasibility, the scheduling of tasks and TFPs must strictly satisfy three *Exclusivity Constraints*:

- Agent exclusivity: An agent can participate in at most one active TFP at any given time. For any two concurrent TFPs sharing an agent, their global temporal intervals $[\mathcal{T}_b, \mathcal{T}_e]$ must not overlap.
- Task exclusivity: A specific task instance can be executed by at most one TFP simultaneously. For any two instances of the same task t_1, t_2 executed by two different TFPs, their respective execution intervals $[\mathcal{T}_b^{t_1}, \mathcal{T}_e^{t_1}]$ and $[\mathcal{T}_b^{t_2}, \mathcal{T}_e^{t_2}]$ must not overlap.
- Team exclusivity: A formed team executes exactly one task at a time. For any two tasks t_1, t_2 executed by the same TFP, their respective execution intervals $[\mathcal{T}_b^{t_1}, \mathcal{T}_e^{t_1}]$ and $[\mathcal{T}_b^{t_2}, \mathcal{T}_e^{t_2}]$ must not overlap.

Consequently, agent's skills are inherently dedicated to a single task at any given moment, naturally preventing the simultaneous use of its skills for two tasks in parallel without requiring an explicit skill exclusivity constraint.

Furthermore, any valid plan must respect *Inherent Constraints* (the internal timing bounds Constr_t within each task) and *Contradictory-Effects Constraints*. Since qualitative skills are static in our model, contradictory effects only apply to fluents and quantitative capacities: two primitive events occurring at the exact same instant τ cannot assign conflicting boolean values to the same fluent $f \in \mathcal{F}$, nor assign diverging absolute values to the same quantitative skill level of an agent.

► **Definition 26** (Team Temporal Plan). Let $\langle \mathcal{M}, I, G \rangle$ be a TTPP. A Team Temporal Plan (TTP) is a tuple $\Pi = \langle \mathcal{P}, \tau \rangle$, where $\mathcal{P} = \{p_1, \dots, p_n\}$ is a finite set of TFPs and τ is a rational-valued scheduling function mapping every primitive event to a specific timepoint. Π is a valid temporal plan for the TTPP if the following conditions are met:

1. *Consistency*: $\mathcal{P}$ strictly satisfies all inherent, contradictory-effect and exclusivity constraints over $\mathcal{M}$.
2. *Execution Validity*: When executing Π from the initial state I (i.e. fluents and quantitative skills are updated at the precise times given by τ):
 - a. For all executed tasks, required conditions $f \in \mathcal{F}$ hold true over their specified requirement intervals.
 - b. For every task t executed by a team C at time τ_t , the team satisfies the Executable Task Conditions (i.e. C covers all required qualitative skills r^{01} and continuously provides sufficient aggregated quantitative capacities r^Q).
 - c. All goal fluents $g \in G$ are true at the end of the execution of Π .

4 Multi-Agent Temporal Planning as SMT

To encode our temporal multi-agent problem into SMT, we must first define the variables representing the timeline of events. Unlike standard SAT encodings that discretize time into fixed steps, our SMT approach uses rational-valued variables to represent exact timepoints. First, we encode the chronological validity of the TFPs. A TFP cannot occur instantaneously, it follows a strict sequence of stages. For any given TFP, its initialization must precede its execution phase (but can not precede the initial timepoint), and finally conclude at its end

time. We also define the *virtual event* $\tau(n \dashrightarrow p)$ corresponding to a past timepoint τ where we establish that agent n is tied to p and can not be used for another TFP. To enforce this structure, we impose the following constraint for all TFPs:

$$\bigwedge_{p \in \mathcal{P}} \left((\mathcal{T}_{Init} - \mathcal{T}_b^p \leq 0) \wedge (\mathcal{T}_b^p - \mathcal{T}_d^p \leq 0) \wedge (\mathcal{T}_d^p - \mathcal{T}_e^p \leq 0) \right)$$

By formulating these temporal relations as difference logic inequalities (e.g. $x - y \leq 0$), we ensure that the SMT solver's arithmetic theory engine can efficiently resolve the temporal bounds without combinatorial explosion.

$$\bigwedge_{\substack{p_1 \in \mathcal{P} \\ n \in N}} \left(\left((n \in C^{p_1}) \Rightarrow \left(\tau(n \dashrightarrow p_1) - \mathcal{T}_b^{p_1} \leq 0 \right) \wedge \left(\bigvee_{\substack{p_2 \in \mathcal{P} \\ p_1 \neq p_2}} \left((n \in C^{p_2}) \wedge (\tau(n \dashrightarrow p_1) - \mathcal{T}_e^{p_2} = 0) \right) \vee \left(\tau(n \dashrightarrow p_1) - \mathcal{T}_{Init} = 0 \right) \right) \right) \right)$$

$$\bigwedge_{\substack{n \in N \\ p_1, p_2 \in \mathcal{P} \\ p_1 \neq p_2}} \left((n \in C^{p_1}) \wedge (n \in C^{p_2}) \Rightarrow \left(\tau(n \dashrightarrow p_1) - \mathcal{T}_e^{p_2} \geq 0 \right) \vee \left(\tau(n \dashrightarrow p_2) - \mathcal{T}_e^{p_1} \geq 0 \right) \right)$$

To track the continuous evolution of quantitative skills, our encoding relies on two mechanisms: local updates via a pass-through logic, and global causal links protected by formulas φ_1 and φ_2 below. On the one hand, the pass-through mechanism acts as a frame axiom. For any task event, if an agent is not part of the assigned team ($\neg(n \in C^p)$), the solver enforces a strict equality between their before and after skill values ($val_{AddB}^{p,t,k,n} = val_{AddA}^{p,t,k,n}$). The event is mathematically transparent to them, preventing unintended resource modifications.

$$\bigwedge_{\substack{p \in \mathcal{P} \\ t \in T}} \left((t \in X^p) \Rightarrow \left(\left(\left(u_{|N|}^{p,t,k} = 0 \right) \wedge \left(u_0^{p,t,k} = v \right) \right) \wedge \bigwedge_{i \in \{1, \dots, |N|\}} \left(u_{i-1}^{p,t,k} - u_i^{p,t,k} - val_{AddA}^{p,t,k,n_{i-1}} + val_{AddB}^{p,t,k,n_{i-1}} = 0 \right) \right) \wedge \left(\left(\neg(n \in C^p) \Rightarrow \left(val_{AddA}^{p,t,k,n} - val_{AddB}^{p,t,k,n} = 0 \right) \right) \wedge \left(val_{AddB}^{p,t,k,n} - val_{AddA}^{p,t,k,n} \leq 0 \right) \wedge \bigwedge_{\substack{n \in N \\ w \in \mathcal{W} \\ n \in N^w \\ k \in K_{qt}^w}} \left(val_{AddA}^{p,t,k,n} - P_{qt}^w(k) \leq 0 \right) \right) \right)$$

On the other hand, for the participating agents, we must link their current before state to a valid past event. This is achieved through virtual timepoints and the two sub-formulas: φ_1 (Causal Support) that forces the solver to find a preceding event (an effect of a past task, or the initial state) whose after value exactly matches the current before requirement, effectively establishing a causal link. φ_2 (Causal Link Protection) that resolves potential threats by ensuring that no other event modifying this agent's specific skill can be scheduled inside the interval between the supporting past event and the current task event. Any interfering task must be constrained to occur before the support or after the current event. As before, we use again a virtual event for each agent n , for example $\tau^{p,n}(t \dashrightarrow k_{qt}(+v))$, to reserve in the past the amount of contribution $val_{AddB}^{p,t,k,n}$ of skill k from agent n for the task t timepoint $\tau^p(t \rightarrow k_{qt}(+v))$.

$$\begin{aligned}
& \bigwedge_{\substack{p \in \mathcal{P} \\ t \in T}} \left((t \in X^p) \Rightarrow \bigwedge_{\substack{n \in N \\ (t \rightarrow k_{qt}(+v)) \in E_t}} (\tau^{p,n}(t \dashrightarrow k_{qt}(+v)) - \tau^p(t \rightarrow k_{qt}(+v)) \leq 0 \wedge \varphi_1 \wedge \varphi_2) \right) \\
\varphi_1 = & \left(\bigvee_{p_2 \in \mathcal{P}} \left(\bigvee_{\substack{t_2 \in T \\ (t_2 \rightarrow k_{qt}(+v_2)) \in E_{t_2} \\ (p \neq p_2 \vee t \neq t_2)}} \left(\begin{aligned} & (t_2 \in X^{p_2}) \\ & \wedge (val_{AddB}^{p,t,k,n} - val_{AddA}^{p_2,t_2,k,n} = 0) \\ & \wedge (\tau^{p,n}(t \dashrightarrow k_{qt}(+v)) - \tau^{p_2}(t_2 \rightarrow k_{qt}(+v_2)) = 0) \end{aligned} \right) \right) \right. \\
& \left. \vee \bigvee_{\substack{t_2 \in T \\ (t_2 \rightarrow k_{qt}(-v_2)) \in E_{t_2}}} \left(\begin{aligned} & (t_2 \in X^{p_2}) \\ & \wedge (val_{AddB}^{p,t,k,n} - val_{DelA}^{p_2,t_2,k,n} = 0) \\ & \wedge (\tau^{p,n}(t \dashrightarrow k_{qt}(+v)) - \tau^{p_2}(t_2 \rightarrow k_{qt}(-v_2)) = 0) \end{aligned} \right) \right) \\
& \left. \vee ((val_{AddB}^{p,t,k,n} = val_{Init}^{k,n}) \wedge (\tau^{p,n}(t \dashrightarrow k_{qt}(+v)) - \mathcal{T}_{Init} = 0)) \right) \\
\varphi_2 = & \bigwedge_{p_3 \in \mathcal{P}} \left((n \in C^{p_3}) \Rightarrow \left(\begin{aligned} & \bigwedge_{\substack{t_3 \in T \\ (t_3 \rightarrow k_{qt}(-v_3)) \in E_{t_3} \\ (p \neq p_3 \vee t \neq t_3)}} \left(\begin{aligned} & (t_3 \in X^{p_3}) \Rightarrow \\ & \tau^{p_3}(t_3 \rightarrow k_{qt}(-v_3)) - \tau^{p,n}(t \dashrightarrow k_{qt}(+v)) \leq 0 \\ & \vee \tau^p(t \rightarrow k_{qt}(+v)) - \tau^{p_3}(t_3 \rightarrow k_{qt}(-v_3)) \leq 0 \end{aligned} \right) \\ & \wedge \bigwedge_{\substack{t_3 \in T \\ (t_3 \rightarrow k_{qt}(+v_3)) \in E_{t_3} \\ (p \neq p_3 \vee t \neq t_3)}} \left(\begin{aligned} & (t_3 \in X^{p_3}) \Rightarrow \\ & \tau^{p_3}(t_3 \rightarrow k_{qt}(+v_3)) - \tau^{p,n}(t \dashrightarrow k_{qt}(+v)) \leq 0 \\ & \vee \tau^p(t \rightarrow k_{qt}(+v)) - \tau^{p_3}(t_3 \rightarrow k_{qt}(+v_3)) \leq 0 \end{aligned} \right) \end{aligned} \right) \right)
\end{aligned}$$

This causal linking mechanism is not limited to quantitative skills, but serves as a universal foundation for our encoding. We apply an identical virtual event formulation to manage the logical consistency of fluents and qualitative skills. For any required propositional condition, the solver instantiates a virtual event that binds the requirement to a valid past establishing event (such as a previous task's effect or the initial state). Simultaneously, the encoding secures this causal chain by ensuring that any conflicting event (e.g. a deletion effect) is strictly isolated outside the protected temporal interval. Consequently, both the numeric and propositional states of the environment remain strictly consistent throughout the timeline. All additional encoding rules are given in Appendix A.

In terms of expressivity, our model captures the numeric and temporal semantics of PDDL2.1, naturally mapping quantitative skills to **increase**, **decrease**, and **assign** effects. While encoding such expressiveness in SAT causes combinatorial explosion through time and resource discretization, our SMT formulation over Quantifier-Free Linear Real Arithmetic (QF-LRA) natively handles continuous variables without artificial discretization. Theoretically, concurrent planning is EXPSpace-complete over discrete time ($\mathbb{T} = \mathbb{Z}$) [12] and generally undecidable over dense time ($\mathbb{T} = \mathbb{Q}$) [10]. However, by bounding the maximum number of TFPs, our encoding restricts the continuous state space to a finite, polynomially bounded set of events. This reduces our specific planning task to a bounded scheduling problem. Since the satisfiability of QF-LRA is NP-complete, our bounded formulation fundamentally lowers the theoretical complexity. To guarantee that our SMT formulation accurately captures the semantics of the Multi-Agent Temporal Planning Problem, we state the following theorem.

► **Theorem 27** (Soundness and Completeness). *Let $\langle \mathcal{M}, I, G \rangle$ be a Team Temporal Planning Problem (TTPP). Any satisfying assignment produced by the SMT encoding corresponds to a valid Team Temporal Plan Π for this TTPP (Soundness). Conversely, if a valid plan Π exists for the TTPP within the bounded number of TFPs, the SMT encoding is satisfiable (Completeness).*

The complete formal proof of this theorem is given in Appendix B.

5 Experimental Evaluation

To evaluate our approach, we model a disaster response scenario, which describes a highly uncertain environment requiring heterogeneous agents to dynamically form teams to execute complex, time-critical tasks (e.g. firefighting, rescuing victims).

5.1 Domain Description and Setup

Our domain defines five innate *qualitative skills* (Search, Rescue, Fire extinguishing, Clear, Mark) and two *quantitative skills* (Water supply and Strength, bounded in $[0, 100]$). To solve the tasks, we define five heterogeneous agent classes, summarized in Table 1.

■ **Table 1** Agent classes and their respective qualitative and quantitative skill profiles.

Class	Type	Qualitative Skills	Quantitative Skills
Firefighter	Human	Fire ext., Rescue, Mark	Water supply, Strength
Reconnaissance Drone	UAV	Search, Mark	-
Firefighting Drone	UAV	Fire ext.	Water supply
Ground Support Drone	UGV	Rescue	Water supply
High-risk Inspection Drone	UGV	Search	Strength

To conduct a robust evaluation, temporal tasks are generated using randomized structural profiles. Tasks can be either *production tasks* (20% probability), establishing 1 to 3 new fluents with no fluent prerequisites, or as *transition tasks* (80%), requiring 1 to $|F|/2$ fluents and establishing 1 to $|F|/2$ new ones, with a 50% chance of consuming each initial condition. Qualitative skills have a 30% probability of being required. Quantitative skills have a 30% chance of being required (sampled uniformly in $[10, 100]$). If a capacity is required, it has a 50% chance of being consumed; otherwise, it has a 5% chance of being restored (sampled in $[10, 100]$). Additionally, dedicated refueling tasks are explicitly generated to replenish specific quantitative capacities. Finally, every task condition and effect is assigned a randomized relative timepoint to accurately simulate complex internal task synchronizations.

5.2 Protocol and metrics

We generated benchmarks varying along three axes: the number of agents $|N| \in [3, 10]$ (uniformly distributed among classes), tasks $|T| \in \{10, 15, 20\}$, and fluents $|F| \in \{6, 12\}$. Since our event-based SMT formulation requires an upper bound on the number of Team Formation Processes (TFPs) to unfold the timeline, we applied a dichotomic search strategy to find the minimal feasible plan. For $|N| \leq 7$, the search explored up to 8 TFPs with a 300s timeout per instance. For $|N| \geq 8$, the bound was extended to 10 TFPs with a 600s timeout.

Instances were encoded into SMT using TouIST [7] and solved with Yices on an Intel Core i7-1370P (32GB RAM). We recorded the translation time (T_{trans}), solving time (T_{solve}), total execution time (T_{exec}), number of teams formed ($|\mathcal{TF}|$), number of executed tasks ($|X|$), and makespan.

It is crucial to note that the **Success Rate** reported in our results strictly isolates non-trivial SAT instances. The remaining percentage encompasses not only classical solver timeouts and proven UNSAT instances, but also trivially solvable scenarios where the goal state was immediately satisfied by the initial state or by uncontrolled environment events (Nature) without requiring any active team formation.

It is important to emphasize why direct empirical comparisons with existing temporal-numeric planners (e.g. OPTIC [2], SMTPlan+ [4] or ARIES [3]) are omitted. Standard PDDL-based and temporal-numeric planners do not natively support dynamic team formation. Attempting to encode our TTP into classical temporal-numeric languages would require explicitly grounding all possible $2^{|N|}$ team combinations as individual actions. This artificial translation induces a combinatorial explosion before the temporal scheduling and resource aggregation begins.

5.3 Results

Tables 2 and 3 summarize the average metrics for successful runs. Preliminary observations indicated that the number of logical fluents exerts a marginal impact compared to the combinatorial explosion of team assignments, thus, our analysis focuses on $|N|$ and $|T|$.

■ **Table 2** Average performance metrics grouped by number of agents ($|N|$). Times in seconds.

$ N $	Success Rate	T_{trans}	T_{solve}	T_{exec}	$ \mathcal{TF} $	$ X $	Makespan
3	29.0%	0.17	2.34	2.51	1.45	4.00	27.64
4	30.7%	0.18	2.75	2.93	1.45	4.28	30.37
5	31.7%	0.20	3.91	4.11	1.20	3.74	25.32
6	32.0%	0.20	5.03	5.23	1.48	4.83	33.99
7	36.0%	0.28	1.20	1.48	1.30	3.57	27.55
8	33.3%	0.49	2.06	2.55	1.30	3.31	21.94
9	26.7%	0.48	6.35	6.84	1.41	3.85	30.07
10	33.0%	0.59	2.64	3.24	1.31	3.60	28.26

■ **Table 3** Average performance metrics grouped by number of tasks ($|T|$). Times in seconds.

$ T $	Success Rate	T_{trans}	T_{solve}	T_{exec}	$ \mathcal{TF} $	$ X $	Makespan
10	27.4%	0.18	2.24	2.43	1.38	3.25	21.05
15	31.0%	0.34	2.96	3.30	1.35	3.95	29.13
20	36.3%	0.42	4.11	4.53	1.35	4.31	32.43

As observed, the SMT encoding demonstrates remarkable efficiency. The translation time (T_{trans}) remains negligible, consistently staying well below one second, which highlights the structural compactness of our TouIST encoding rules.

The solving time (T_{solve}) is highly competitive. Across all configurations, the solver successfully navigates the complex interplay between dynamic capability matching and dense temporal constraints in an average of 2 to 6 seconds. While scaling the problem naturally induces combinatorial growth, the framework comfortably manages up to 10 heterogeneous agents and 20 temporal tasks within acceptable planning windows.

Furthermore, analyzing the generated plans reveals an interesting structural behavior: the average number of formed teams ($|\mathcal{TF}| \approx 1.3$) is above 1 which means that, on average, several TFPs are created, but the fact that the average number is between 1 and 2 also indicates that the solver actively leverages the agents' combined skills by packing multiple sequential tasks ($|X| \approx 4$) into a single, long-lived team rather than frequently dissolving and reassembling different groups. This behavior is fundamentally tied to the nature of standard

SMT solvers, which are designed to find the first logically valid assignment (satisfiability) rather than the globally optimal one. Without an explicit objective function to minimize the makespan, the solver lacks the mathematical incentive to parallelize tasks across multiple simultaneous teams. Extending this framework to support optimization metrics to actively minimize the makespan represents a highly promising avenue for future work.

5.4 Discussion on Expressive Capabilities

To clarify the contribution of our model independently of the underlying solver, it is essential to position its expressive capabilities against existing temporal planning frameworks such as PDDL2.1. While standard action-based languages are highly expressive and could theoretically model dynamic team formation, doing so requires severe and inefficient structural workarounds. To represent a cooperative task, a modeler would have to either explicitly ground a distinct action for every possible subset of agents (causing an immediate $2^{|N|}$ combinatorial explosion in the domain definition itself), or introduce intricate sequences of intermediate “join-team” and “leave-team” actions. The latter approach artificially inflates the plan length and severely complicates the temporal synchronization of concurrent effects.

Our framework offers a fundamentally different representation that avoids these pitfalls by treating dynamic team formation and capability aggregation as first-order concepts. The expressive advantage of our model does not lie in bypassing classical decidability limits, but in its ability to natively and compactly represent cooperative task execution: a task dynamically dictates its qualitative and quantitative requirements over time, and a valid team emerges to fulfill them simultaneously. This provides a significantly more natural representation for complex Multi-Robot Task Allocation (MRTA) scenarios, by clearly decoupling the temporal logic of the tasks from the combinatorial complexity of agent assignments.

6 Conclusion and Perspectives

In this paper, we have introduced a new framework for multi-agent temporal planning that effectively integrates dynamic team formation with event-based temporal reasoning. Our approach overcomes the traditional limitations of SAT encoding for complex timelines and also unifies the dynamics of team formation with the arithmetic complexity of quantitative skill allocation through SMT encoding. Our experiments on disaster response scenarios demonstrate that our unified encoding is both expressive and computationally scalable, successfully coordinating heterogeneous agents to execute complex and distributed tasks. This work establishes a robust foundation for multi-agent temporal planning where cooperative team structures are not predefined, but emerge dynamically as a direct result of the temporal scheduling process.

In this article, we focus on a framework describing only two classes of skills, with fixed domains and aggregation functions. Future work may include the generalization of classes, with a higher-level definition of their domains and aggregation functions. Thus, this would enable us to use our model to define a wider range of problems in a flexible manner.

The experiments presented in this work focused on finding valid plans, while optimization was performed manually by adjusting the number of TFPs in order to identify the minimum achievable makespan for a given problem instance. However, some TFPs may be created and certain tasks executed even though they are not strictly necessary for the plan to be valid. A promising direction for future work would be to investigate multi-objective optimization approaches that jointly consider the makespan, the number of TFPs, and the number of executed tasks. Such optimization could notably be addressed using another solver such as Z3 through the SMT-LIB formalism.

References

- 1 Clark Barrett and Cesare Tinelli. Satisfiability modulo theories. In *Handbook of model checking*, pages 305–343. Springer, 2018. doi:10.1007/978-3-319-10575-8_11.
- 2 J. Benton, Amanda Jane Coles, and Andrew Coles. Temporal planning with preferences and time-dependent continuous costs. In Lee McCluskey, Brian Charles Williams, José Reinaldo Silva, and Blai Bonet, editors, *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012, Atibaia, São Paulo, Brazil, June 25-19, 2012*. AAAI, 2012. URL: <http://www.aaai.org/ocs/index.php/ICAPS/ICAPS12/paper/view/4699>.
- 3 Arthur Bit-Monnot. Experimenting with Lifted Plan-Space Planning as Scheduling: Aries in the 2023 IPC. In *Proceedings of the 2023 International Planning Competition*, Prague, Czech Republic, July 2023.
- 4 Michael Cashmore, Maria Fox, Derek Long, and Daniele Magazzeni. A compilation of the full PDDL+ language into SMT. In *Proceedings of the international conference on automated planning and scheduling*, volume 26, pages 79–87, 2016. URL: <http://www.aaai.org/ocs/index.php/ICAPS/ICAPS16/paper/view/13101>.
- 5 Luiz Chaimowicz, Mario F.M. Campos, and Vijay Kumar. Dynamic role assignment for cooperative robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 293–298. IEEE, 2002.
- 6 Martin C. Cooper, Frederic Maris, and Pierre Régnier. Monotone temporal planning: Tractability, extensions and applications. *J. Artif. Intell. Res. (JAIR)*, 50:447–485, 2014. doi:10.1613/JAIR.4358.
- 7 Jorge H. Fernandez, Olivier Gasquet, Andreas Herzig, Dominique Longin, Emiliano Lorini, Frédéric Maris, and Pierre Régnier. Touist: a friendly language for propositional logic and more. In *29th International Joint Conference on Artificial Intelligence (IJCAI 2020)*, pages 5240–5242. International Joint Conferences on Artificial Intelligence Organization, 2020. doi:10.24963/IJCAI.2020/756.
- 8 Matthew E. Gaston and Marie DesJardins. Agent-organized networks for dynamic team formation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 230–237, 2005. doi:10.1145/1082473.1082508.
- 9 Brian P. Gerkey and Maja J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9):939–954, 2004. doi:10.1177/0278364904045564.
- 10 Nicola Gigante, Andrea Micheli, Angelo Montanari, and Enrico Scala. Decidability and complexity of action-based temporal planning over dense time. *Artif. Intell.*, 307:103686, 2022. doi:10.1016/J.ARTINT.2022.103686.
- 11 Tyler Gunn and John Anderson. Dynamic heterogeneous team formation for robotic urban search and rescue. *Journal of Computer and System Sciences*, 81(3):553–567, 2015. doi:10.1016/J.JCSS.2014.11.009.
- 12 Jussi Rintanen. Complexity of concurrent temporal planning. In Mark S. Boddy, Maria Fox, and Sylvie Thiébaux, editors, *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling, ICAPS 2007, Providence, Rhode Island, USA, September 22-26, 2007*, pages 280–287. AAAI, 2007. URL: <http://www.aaai.org/Library/ICAPS/2007/icaps07-036.php>.
- 13 Jussi Rintanen. Discretization of temporal models with application to planning with SMT. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.
- 14 Jussi Rintanen. Planning and SAT. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 765–789. IOS Press, 2021. doi:10.3233/FAIA201003.
- 15 Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial intelligence*, 101(1-2):165–200, 1998. doi:10.1016/S0004-3702(98)00045-9.
- 16 Milind Tambe. Towards flexible teamwork. *Journal of artificial intelligence research*, 7:83–124, 1997. doi:10.1613/JAIR.433.

A Additional Encoding Rules (Complete Encoding)

This section details the exhaustive SMT encoding rules applied to generate the valid Team Temporal Plans. To ensure readability, formulas use the mathematical notations introduced in Section 3 and Section 4. Let $\tau^p(t)$ denote the base reference timepoint of task t within the TFP p . $\mathcal{T}_b^{p,t}$ and $\mathcal{T}_e^{p,t}$ represent the specific start and end temporal execution boundaries of task t within p . For quantitative aggregations, $u_i^{p,t,k}$ represents the cumulative sum variable over the agents.

Structural Constraints

Agents cannot belong to a Team Formation Process (TFP) if it does not execute any task:

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{n \in N} \left((n \in C^p) \Rightarrow \bigvee_{t \in T} (t \in X^p) \right)$$

All TFPs must strictly start after the initial chronological timepoint $\mathcal{T}_{Init}$:

$$\bigwedge_{p \in \mathcal{P}} \left(\mathcal{T}_{Init} - \mathcal{T}_b^p \leq 0 \right)$$

If a task t is executed within a TFP p , its precise temporal bounds $\mathcal{T}_b^{p,t}$ and $\mathcal{T}_e^{p,t}$ are restrained within the TFP execution phase $[\mathcal{T}_d^p, \mathcal{T}_e^p]$. Furthermore, every primitive event $e \in E_t$ must strictly occur within the task boundaries:

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{t \in T} \left((t \in X^p) \Rightarrow \left((\mathcal{T}_d^p - \mathcal{T}_b^{p,t} \leq 0) \wedge (\mathcal{T}_e^p - \mathcal{T}_e^{p,t} \geq 0) \right) \wedge \bigwedge_{e \in E_t} \left((\mathcal{T}_b^{p,t} - \tau^p(e) \leq 0) \wedge (\mathcal{T}_e^{p,t} - \tau^p(e) \geq 0) \right) \right)$$

Inherent Internal Task Constraints ($Constr_t$)

For every executed task t in a TFP p , the relative occurrence of its condition and effect events must respect the structural offset bounds Δ_t defined in its domain profile:

Fluents Constraints:

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{t \in T} \left(\begin{array}{l} \bigwedge_{(f \rightarrow t) \in E_t} \left(\tau^p(f \rightarrow t) - \tau^p(t) = \Delta_t(f \rightarrow t) \right) \\ \wedge \bigwedge_{(t \rightarrow f) \in E_t} \left(\tau^p(t \rightarrow f) - \tau^p(t) = \Delta_t(t \rightarrow f) \right) \\ \wedge \bigwedge_{(t \rightarrow \neg f) \in E_t} \left(\tau^p(t \rightarrow \neg f) - \tau^p(t) = \Delta_t(t \rightarrow \neg f) \right) \end{array} \right)$$

Quantitative Skills Constraints:

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{t \in T} \left(\begin{array}{l} \bigwedge_{((k_{qt} \geq v) \rightarrow t) \in E_t} \left(\tau^p((k_{qt} \geq v) \rightarrow t) - \tau^p(t) = \Delta_t((k_{qt} \geq v) \rightarrow t) \right) \\ \wedge \bigwedge_{(t \rightarrow k_{qt}(+v)) \in E_t} \left(\tau^p(t \rightarrow k_{qt}(+v)) - \tau^p(t) = \Delta_t(t \rightarrow k_{qt}(+v)) \right) \\ \wedge \bigwedge_{(t \rightarrow k_{qt}(-v)) \in E_t} \left(\tau^p(t \rightarrow k_{qt}(-v)) - \tau^p(t) = \Delta_t(t \rightarrow k_{qt}(-v)) \right) \end{array} \right)$$

Qualitative Skills Constraints:

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{t \in T} \bigwedge_{(k_{ql} \rightarrow t) \in E_t} \left(\tau^p(k_{ql} \rightarrow t) - \tau^p(t) = \Delta_t(k_{ql} \rightarrow t) \right)$$

Qualitative Capability Checking

Tasks requiring qualitative skills k_{ql} must be executed by a team where at least one assigned agent natively masters the specific skill:

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{t \in T} \left((t \in X^p) \Rightarrow \bigwedge_{(k_{ql} \rightarrow t) \in E_t} \bigvee_{\substack{n \in N \\ k_{ql} \in m_n^{01}}} (n \in C^p) \right)$$

Causality and Protection over Propositional Fluents (F)

Goal Fluents (G): For each goal $f \in G$, its virtual requirement event $\tau^{req}(G, f)$ must occur at or before $\mathcal{T}_{Goal}$. It must be supported by a preceding production effect or the initial state I . Any deletion of f must explicitly occur strictly before the protection interval starts.

$$\bigwedge_{f \in G} \left(\begin{array}{l} \left(\tau^{req}(G, f) - \mathcal{T}_{Goal} \leq 0 \right) \\ \wedge \left(\bigvee_{p_1 \in \mathcal{P}} \bigvee_{\substack{t_1 \in T \\ (t_1 \rightarrow f) \in E_{t_1}}} \left((t_1 \in X^{p_1}) \wedge \left(\tau^{req}(G, f) - \tau^{p_1}(t_1 \rightarrow f) = 0 \right) \right) \right) \\ \vee \left\{ \begin{array}{ll} \left(\tau^{req}(G, f) - \mathcal{T}_{Init} = 0 \right) & \text{if } f \in I \\ \perp & \text{otherwise} \end{array} \right. \\ \wedge \bigwedge_{p_2 \in \mathcal{P}} \bigwedge_{\substack{t_2 \in T \\ (t_2 \rightarrow \neg f) \in E_{t_2}}} \left((t_2 \in X^{p_2}) \Rightarrow \tau^{p_2}(t_2 \rightarrow \neg f) - \tau^{req}(G, f) < 0 \right) \end{array} \right)$$

Task Fluents Conditions ($CondF$): Any task condition f must similarly be supported by a valid past event. Any conflicting deletion must be scheduled entirely outside the resulting causal protection link.

$$\bigwedge_{p_2 \in \mathcal{P}} \bigwedge_{t_2 \in T} \left((t_2 \in X^{p_2}) \Rightarrow \begin{array}{l} \left(\tau^{p_2, req}(f \dashrightarrow t_2) - \tau^{p_2}(f \rightarrow t_2) \leq 0 \right) \\ \wedge \left(\bigvee_{p_1 \in \mathcal{P}} \bigvee_{\substack{t_1 \in T \\ (t_1 \rightarrow f) \in E_{t_1}}} \left((t_1 \in X^{p_1}) \wedge \left(\tau^{p_2, req}(f \dashrightarrow t_2) - \tau^{p_1}(t_1 \rightarrow f) = 0 \right) \right) \right) \\ \vee \left\{ \begin{array}{ll} \left(\tau^{p_2, req}(f \dashrightarrow t_2) - \mathcal{T}_{Init} = 0 \right) & \text{if } f \in I \\ \perp & \text{otherwise} \end{array} \right. \\ \wedge \bigwedge_{p_3 \in \mathcal{P}} \bigwedge_{\substack{t_3 \in T \\ (t_3 \rightarrow \neg f) \in E_{t_3}}} \left(\begin{array}{l} (t_3 \in X^{p_3}) \Rightarrow \\ \left(\tau^{p_3}(t_3 \rightarrow \neg f) - \tau^{p_2, req}(f \dashrightarrow t_2) < 0 \right) \\ \vee \left(\tau^{p_2}(f \rightarrow t_2) - \tau^{p_3}(t_3 \rightarrow \neg f) < 0 \right) \end{array} \right) \end{array} \right)$$

Quantitative Constraints and Causal Link Logic

The encoding of quantitative skills relies on two symmetrical mechanisms for each event type (Addition, Consumption, and Condition): the intra-task capacity aggregation, and the inter-task causal link protection. As the complete formulas for capacity production ($AddQt$)

are already detailed in Section 4, we define the consumption (*DelQt*) and requirement (*CondQt*) constraints by highlighting their specific deviations from the *AddQt* base formulas to maintain brevity.

1. Task Quantitative Aggregation: The aggregation logic iterates over all agents to sum their individual contributions $u_i^{p,t,k}$.
 - For Requirements (*CondQt*), the exact matching $u_0^{p,t,k} = v$ is relaxed to a lower bound $u_0^{p,t,k} \geq v$. Furthermore, since a condition does not consume the resource, the before/after state variables (val_{AddA}, val_{AddB}) are replaced by a single contribution variable $val_{Cond}^{p,t,k,n}$. For non-participating agents, we enforce $val_{Cond} = 0$.
 - For Consumptions (*DelQt*), the logic mirrors *AddQt* but reverses the before/after dynamics. The sequence enforces $u_{i-1}^{p,t,k} - u_i^{p,t,k} - val_{DelB}^{p,t,k,n} + val_{DelA}^{p,t,k,n} = 0$. For participating agents, we ensure non-negativity and strict value tracking: $val_{DelB} - val_{DelA} \geq 0$ and $val_{DelA} \geq 0$. Non-participating agents pass their values through unchanged ($val_{DelB} = val_{DelA}$).
2. Virtual Causal Events and Protection: To guarantee causal consistency, every individual contribution ($val_{Cond}^{p,t,k,n}$ or $val_{DelB}^{p,t,k,n}$) must be anchored to a past establishing event without any contradictory events occurring in between. The formulas are strictly identical to the φ_1 (Causal Support) and φ_2 (Causal Protection) formulas provided for *AddQt* in Section 4, subject to the following variable substitutions:
 - For *CondQt*: The virtual timepoint $\tau^{p,n}(t \dashrightarrow k_{qt}(+v))$ is replaced by a condition virtual timepoint (e.g. $\tau^{p,n}((k_{qt} \geq v) \dashrightarrow t)$), and the target variable $val_{AddB}^{p,t,k,n}$ is replaced by $val_{Cond}^{p,t,k,n}$.
 - For *DelQt*: The virtual timepoint $\tau^{p,n}(t \dashrightarrow k_{qt}(+v))$ is replaced by $\tau^{p,n}(t \dashrightarrow k_{qt}(-v))$, and the target variable $val_{AddB}^{p,t,k,n}$ is replaced by $val_{DelB}^{p,t,k,n}$.

Exclusivity Constraints (Mutex)

Agent Exclusivity: An agent n assigned to two distinct TFPs ensures strict sequentiality between the deployment intervals.

$$\bigwedge_{n \in N} \bigwedge_{\substack{p_1 \in \mathcal{P} \\ p_2 \in \mathcal{P} \\ p_1 \neq p_2}} \bigwedge_{p_2 \in \mathcal{P}} \left((n \in C^{p_1}) \wedge (n \in C^{p_2}) \Rightarrow (\mathcal{T}_b^{p_1} - \mathcal{T}_e^{p_2} \geq 0) \vee (\mathcal{T}_b^{p_2} - \mathcal{T}_e^{p_1} \geq 0) \right)$$

Task Exclusivity: Instances of the same task cannot be concurrently executed across distinct TFPs.

$$\bigwedge_{p_1 \in \mathcal{P}} \bigwedge_{\substack{p_2 \in \mathcal{P} \\ p_1 \neq p_2}} \bigwedge_{t \in T} \left((t \in X^{p_1}) \wedge (t \in X^{p_2}) \Rightarrow (\mathcal{T}_b^{p_1,t} - \mathcal{T}_e^{p_2,t} \geq 0) \vee (\mathcal{T}_b^{p_2,t} - \mathcal{T}_e^{p_1,t} \geq 0) \right)$$

Team Exclusivity: Tasks executed consecutively within the same TFP must not internally overlap.

$$\bigwedge_{p \in \mathcal{P}} \bigwedge_{\substack{t_1 \in T \\ t_2 \in T \\ t_1 \neq t_2}} \left((t_1 \in X^p) \wedge (t_2 \in X^p) \Rightarrow (\mathcal{T}_b^{p,t_1} - \mathcal{T}_e^{p,t_2} \geq 0) \vee (\mathcal{T}_b^{p,t_2} - \mathcal{T}_e^{p,t_1} \geq 0) \right)$$

B Proof of the Soundness and Completeness Theorem

In this section, we provide the formal proof of Theorem 27 establishing the equivalence between the satisfiability of our SMT encoding and the existence of a valid Team Temporal Plan for a given Team Temporal Planning Problem (TTPP).

► **Theorem 27** (Soundness and Completeness). *Let $\langle \mathcal{M}, I, G \rangle$ be a Team Temporal Planning Problem (TTPP). Any satisfying assignment produced by the SMT encoding corresponds to a valid Team Temporal Plan Π for this TTPP (Soundness). Conversely, if a valid plan Π exists for the TTPP within the bounded number of TFPs, the SMT encoding is satisfiable (Completeness).*

Proof. The proof is divided into two parts: establishing soundness by formally extracting a valid plan from a satisfying SMT assignment, and establishing completeness by mapping a valid plan back to a satisfying assignment.

Soundness. Let $\mathcal{I}$ be a satisfying assignment (a valid model) returned by the SMT solver for the encoded formula over the QF-LRA theory, generated for the specific TTPP $\langle \mathcal{M}, I, G \rangle$. We construct a Team Temporal Plan $\Pi = \langle \mathcal{P}, \tau \rangle$ using the extraction mapping defined in Section A. We must prove that Π satisfies all conditions of a valid temporal plan for this TTPP (Definition 26):

- *Structural Consistency and Exclusivity:* Because $\mathcal{I}$ strictly satisfies the structural and mutex difference-logic constraints, all task bounds $[\mathcal{T}_b^{p,t}, \mathcal{T}_e^{p,t}]$ are appropriately ordered and enclosed within their respective TFP deployment phases. Furthermore, for any two active TFPs sharing an agent or executing the same task, the temporal intervals are strictly disjoint. Thus, Π is physically plausible and respects all exclusivity constraints.
- *Inherent Constraints:* The SMT formulation enforces precise arithmetic offsets $\tau^p(e) - \tau^p(t) = \Delta_t(e)$. Consequently, the relative timing of all primitive events within every scheduled task in Π exactly matches the domain definition Constr_t .
- *Execution Validity and Causal Consistency:* The pass-through logic (e.g., $u_{i-1}^{p,t,k} - u_i^{p,t,k} - \text{val}_{\text{Cond}} = 0$) mathematically guarantees that the total capacity required by a task is exactly the sum of the individual contributions of the assigned agents in Π . Crucially, for any temporal requirement (whether a propositional fluent or a quantitative resource) scheduled at τ_{req} , the causal support formula φ_1 forces $\mathcal{I}$ to anchor a virtual support event to a strictly preceding establishing event (or to the initial state I). Simultaneously, the protection formula φ_2 logically isolates the temporal interval $[\tau_{\text{sup}}, \tau_{\text{req}}]$. It guarantees that any potentially contradictory event τ_{conf} is scheduled such that either $\tau_{\text{conf}} < \tau_{\text{sup}}$ or $\tau_{\text{conf}} > \tau_{\text{req}}$. Thus, no Contradictory-Effects occur, and Π accurately satisfies the Executable Task Conditions.
- *Initial and Goal States Consistency:* The SMT encoding explicitly roots the causal chains in the initial state I , meaning the execution of Π originates from a valid configuration. Finally, the goal constraints enforce that all $g \in G$ are supported by a valid establishing event prior to $\mathcal{T}_{\text{Goal}}$ and protected from subsequent deletions, ensuring the final state reached by Π correctly satisfies G .

Since all criteria (consistency, execution validity, initial state consistency, and goal reachability) are met, Π is a valid Team Temporal Plan for the given TTPP.

Completeness. Let $\Pi = \langle \mathcal{P}, \tau \rangle$ be a valid Team Temporal Plan for the TTPP $\langle \mathcal{M}, I, G \rangle$, operating within the established bound n_{cfp} . We construct an assignment $\mathcal{I}$ for the SMT formula as follows:

1. Assign the boolean variables ($t \in X^p$) and ($n \in C^p$) to true according to the exact sets X and C defined in $\mathcal{P}$. Assign all unused TFP variables to false.
2. Assign the temporal variables ($\mathcal{T}_b^p, \tau^p(t), \tau^p(e)$) to the exact rational values dictated by the scheduling function τ mapping the events in Π .

3. Assign the quantitative capability variables (val_{AddA} , val_{DelB} , etc.) to the numeric values representing the agents' true resource levels at those specific timepoints during the valid execution of Π .

We now demonstrate that $\mathcal{I}$ satisfies every clause of the encoded formula:

- Because Π is a valid TTP, it inherently respects all chronological boundaries and internal task durations. Thus, all corresponding structural difference-logic equations ($\leq, =$) naturally evaluate to true in $\mathcal{I}$.
- Because Π rigorously prevents agent and task overlapping, the continuous real-time intervals in τ are fully disjoint. Therefore, the logical disjunctions within the Mutex constraints evaluate to true.
- Because Π is validly executable from the initial state I , all required conditions and resources are correctly provided when needed. Consequently, every condition possesses a valid establishing event in the past timeline of Π . By assigning the virtual event variables ($\tau^{p,n}(t \dashrightarrow k_{qt}(+v))$, τ^{req} , etc.) to the exact timepoints of these past supportive events, the equality constraints within the causal support formulas (φ_1) evaluate to true.
- Furthermore, because Π contains zero unresolvable contradictory effects, strict temporal separation exists between any support event and any interfering effect. Thus, the temporal boundaries inherently satisfy the strict disjunctions required by the protection intervals (φ_2).
- Finally, since Π successfully reaches the goal state G , the goal-specific causal link formulas accurately map to a final sequence of valid establishing events.

Since the assignment $\mathcal{I}$ flawlessly maps the valid properties of the continuous temporal plan into the QF-LRA domains without discretization artifacts, the entire logical formula evaluates to true, thus proving the completeness of the encoding. ◀