Strong, Weak, and Dynamic Consistency in Fuzzy Conditional Temporal Problems

M. Falda and **F. Rossi** and **K. B. Venable**
Dept. of Pure and Applied Mathematics
University of Padova, Italy

Abstract

Conditional Temporal Problems (CTPs) allow for the representation of temporal and conditional plans, dealing simultaneously with uncertainty and temporal constraints. In this paper, CTPs are generalized to CTPPs by adding preferences to the temporal constraints and by allowing fuzzy thresholds for the occurrence of some events. The usual consistency notions (strong, weak and dynamic) are then extended to encompass the new setting, and their corresponding testing algorithms are provided. We show that the complexity of the algorithms does not increase w.r.t. their classical counterparts for CTPs. We also show that our framework generalizes STPPUs as well, another temporal framework with uncertainty and preferences. This means that controllability in STPPUs can be translated to consistency in CTPPs, indicating a strong theoretical connection among the two formalisms.

Introduction

Many systems and applications need to be able to reason with alternative situations, plans, contexts and to know what holds in each of them. Moreover, they may have to set temporal constraints on events and actions. Conditional Temporal Problems (CTPs) (Tsamardinos, Vidal, & Pollack 2003) are a formalism that allows for modeling conditional and temporal plans which deal with the uncertainty arising from the outcome of observations and with complex temporal constraints. In CTPs the usual notion of consistency is replaced by three notions, weak, strong and dynamic consistency, which differ on the assumptions made on the knowledge available.

Another class of temporal reasoning problems that deals with similar scenarios are Simple Temporal Problems with Uncertainty (STPUs) (Vidal & Fargier 1999). In such problems the uncertainty lies in the lack of control the agent has over the time at which some events occur. Such events are said to be controlled by "Nature". In STPUs consistency is called controllability and, similarly to CTPs, there are three notions, weak, strong and dynamic controllability, based on different assumptions made on the uncontrollable variables. Despite the fact that consistency in CTPs and controllability in STPUs appear similar, their relation has not been formally investigated.

Furthermore, in rich application domains it is often necessary to handle not only temporal constraints and conditions, but also preferences over the execution of actions. Preferences have been added to STPUs in (Rossi, Venable, & Yorke-Smith 2006); in addition to expressing uncertainty, in STPPUs contingent constraints can be soft, meaning that different preference levels are associated to different durations of events.

In this paper we introduce the CTPP model, an extension of CTPs which adds preferences to the temporal constraints and generalizes the simple Boolean conditions to fuzzy rules; these rules activate the occurrence of some events on the basis of fuzzy thresholds. Moreover, also the activation of the events is characterized by a preference function over the domain of the event. This provides an additional gain in expressiveness, allowing one to model the dynamic aspect of preferences that change over time.

Quantitative temporal constraint problems have been used for many applications in practice, ranging from space applications (MAPGEN (Ai-Chang *et al.* 2004)) to temporal databases (Combi & Pozzi 2006) and personal assistance (Autominder, (Pollack *et al.* 2003)). We expect CTPPs to be useful in all of the above.

After defining CTPs with fuzzy preferences, we extend all the consistency notions of CTPs. Moreover, we provide algorithms for testing such new notions which are in the same complexity class as their classical counterparts. Finally, we show how the STPPUs are related to CTPPs by providing a mapping from STPPUs to CTPPs (and thus also from STPUs to CTPs) which preserves the controllability/consistency notions. In particular, such a mapping proves that CTPPs are a more expressive model. All proofs have been omitted for lack of space.

Background

STPs and STPPs. A Simple Temporal Problem (STP) (Dechter, Meiri, & Pearl 1991) is defined as a set of variables V, each of which corresponds to an instantaneous event, and a set E of constraints between the variables. The constraints are binary and are of the form $l_{ij} \leq x_i - x_j \leq u_{ij}$ with $x_i, x_j \in V$ and $l_{ij}, u_{ij} \in \Re$; l_{ij} and u_{ij} are called the bounds of the constraint.

Preferences have been introduced in STPs by (Khatib *et al.* 2001), defining Simple Temporal Problems with Pref-

erences (STPPs). In particular, a soft temporal constraint $< I, f >$ is specified by means of a preference function on the interval, $f : I \rightarrow [0, 1]$, where $I = [l_{ij}, u_{ij}]$. An STPP is said to be consistent with preference degree α if there exists an assignment of its variables that satisfies all constraints and that has preference α. The preference of an assignment is obtained by taking the $minimum$ of the preferences given by each constraint to the projection of the assignment onto its variables. An optimal solution is one such that there is no other solution with higher preference. Such a solution ca be found in polynomial time (Khatib *et al.* 2001).

STPUs and STPPUs. STPUs (Vidal & Fargier 1999) are STPs in which the temporal constraints are divided in two classes: those representing durations under the control of the agent (called requirement constraints) and those representing durations decided by "Nature" (called contingent constraints). Such a partition induces a similar partition over the variables. In (Rossi, Venable, & Yorke-Smith 2006) STPUs are extended to preferences by replacing STP constraints with soft temporal constraints. Thus an STPPU is a tuple $< N_e, N_c, L_r, L_c >$ where N_e is the set of executable timepoints, N_c is the set of contingent timepoints, L_r is a set of soft requirement constraints, and L_c is a set of soft contingent constraints. The notions of controllability of STPUs are extended to handle preferences. Here we focus on two of such notions. An STPPU is said to be α-strongly controllable is there is a fixed way to assign the values to the variables in N_e such that whatever Nature will choose for the variables in N_c the resulting assignment is either optimal (if Nature's choice prevents from achieving preference level α) or it has preference α. Optimal weak controllability simply requires the existence of an optimal way to assign values to the variables in N_e given an any assignment to those in N_c.

CTPs. CTPs (Tsamardinos, Vidal, & Pollack 2003) extend temporal constraint satisfaction problems (Dechter, Meiri, & Pearl 1991) by adding observation variables and by conditioning the occurrence of some events on the presence of some properties of the environment. A CTP is a tuple $< V, E, L, OV, O, \mathcal{P} >$ where $\mathcal{P}$ is a set of Boolean atomic propositions, V is a set of variables, E is a set of temporal constraints between pairs of variables in V, $L : V \rightarrow \mathcal{Q}^*$ is a function attaching conjunctions of literals in $\mathcal{Q} = \{p_i : p_i \in P\} \cup \{\neg p_i : p_i \in P\}$ to each variable in V, $OV \subseteq V$ is the set of observation variables, and $O : \mathcal{P} \rightarrow OV$ is a bijective function that associates an observation variable to a proposition. The observation variable $O(A)$ provides the truth value for A. In V there is usually a variable denoting the origin time, set to 0. In this paper this variable will be denoted by x_0. Thus, in CTPs, variables are labelled with conjunctions of literals, and the truth value of such labels are used to determine whether variables represent events that are part of the temporal problem. In this paper we consider only CTPs where E contains only STP constraints. In a CTP, for a variable to be executed, its associated label must be true. The truth values of the propositions appearing in the labels are provided when the corresponding observation variables are executed. The constraint graph of a CTP is a graph where nodes correspond to variables and edges to constraints. Nodes v is labeled with $L(v)$ and edge c is labeled with the interval of constraint c. Labels equal to $true$ are not specified. An *execution scenario* s is a conjunction of literals that partitions the set of variables in two subsets: the subset of the variables that will be executed because their label is true given s, and the subset of the other variables, that will not be executed. SC is the set of all scenarios. Given a scenario s, its *projection*, $Pr(s)$, is the set of variables that are executed under s and all the constraints between pairs of them. $Pr(s)$ is a non-conditional temporal problem.

Figure 1 shows an example inspired from (Tsamardinos, Vidal, & Pollack 2003). The example is about a plan to go skiing at station $Sk1$ or $Sk2$, depending on the condition of road R. Station $Sk2$ can be reached in any case, while station $Sk1$ can be reached only if road R is accessible. If $Sk1$ is reachable, we choose to go there. Moreover, temporal constraints limit the arrival times at the skiing station. The condition of road R can be assessed when arriving at village W. In the figure, variables XY_s and XY_e represent the start and the end time for the trip from X to Y. Node $O(A)$, where A = "road R is accessible" is HW_e. There are two scenarios, A on variables $\{x_0, HW_s, HW_e, WSk1_s, WSk1_e\}$ and $\neg A$ on variables $\{x_0, HW_s, HW_e, WSk2_s, WSk2_e\}$.

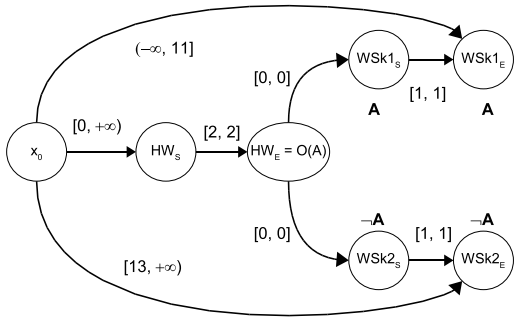

Figure 1: Example of Conditional Temporal Problem.

In CTPs there are three different notions of consistency depending on the assumptions made about the availability of observation information:

- *Strong Consistency (SC).* Strong consistency applies when no information is available. A CTP is strongly consistent if there is a fixed way to assign values to all the variables so that all constraints are satisfied independently of the observations. A CTP is strongly consistent if and only if its non-conditional counterpart is consistent. Therefore, an algorithm to check SC of a CTP takes the same time as checking the consistency of an STP, which is polynomial.
- *Weak Consistency (WC).* Weak consistency applies when all information is available before execution. A CTP is weakly consistent if the projection of any scenario is consistent. Checking WC is a co-NP complete problem

(Tsamardinos, Vidal, & Pollack 2003). A brute force algorithm to check WC can check the consistency of all projections, possibly exploiting equivalent scenarios and shared paths.

- *Dynamic Consistency (DC).* Dynamic consistency (DC) assumes that information about observations becomes known during execution. A CTP is dynamically consistent if it can be executed so that the current partial solution can be consistently extended independently of the upcoming observations.

The CTP depicted in Figure 1 is not DC. In fact, if A is true then we have to leave home after 10, if A is false we have to leave home before 8. However, being at village W is a precondition for the observation of proposition A and this fact prevents us to observe A before leaving home. Therefore we cannot distinguish between the two scenarios A and $\neg A$ in time to schedule our departure from home accordingly.

Fuzzifying CTPs

The conditional nature of CTPs is enclosed in the variables' labels, whose truth value enables or disables the presence of variables in the problem. Such labels indeed act as rules that select different execution paths, which, given variable v and its label $L(v)$, can be written as follows: IF $L(v)$ THEN EXECUTE (v).

The idea of fuzzifying such kind of rules has been already taken into consideration, for example in the field of fuzzy control (Lee 1990; Cox 1992). In fact, real world objects often do not present a crisp membership and classical Logics has difficulties to describe some concepts (e.g. "tall", "young", et.c.). Another problem is that temporal information is often affected by imprecision or vagueness.

In a general study of such rules (Dubois & Prade 1996), both the premise and the consequence of the rule have been equipped with truth degrees associated with them. We will do the same for CTP's rules.

In our case, however, these two degrees have different meanings: the degree of the premise is used to establish if the variable should be executed, and therefore provides a truth value; the degree of the consequence, instead, can be considered as a preference on the execution of the variable.

Boolean propositions were justified in CTPs, where labels were evaluated in a crisp way, but in CTPPs they would reduce the expressiveness of the fuzzy rules; for this reason CTPPs will be equipped with a set $\mathcal{P}$ of fuzzy atomic propositions and a set of fuzzy literals $\mathcal{Q} = \{p_i : p_i \in \mathcal{P}\} \cup \{\neg p_i : p_i \in \mathcal{P}\}$ which are mapped to values from $[0, 1]$ by an interpretation function.

Definition 1 (Interpretation function)**.** An interpretation function is a function $deg : \mathcal{W} \subseteq \mathcal{Q} \rightarrow [0, 1]$, where $l \in \mathcal{W}$ iff $\neg l \in \mathcal{W}$ and $\forall l \in \mathcal{W}, deg(\neg l) = 1 - deg(l)$.

The rules we will use to fuzzify CTPs are of the form

$$\text{IF } pt(L(v), deg) > \alpha \text{ THEN EXECUTE } (v) : cp(minL(v, deg))$$

where $L(v) \in \mathcal{Q}^*$ is the "fuzzy" label of variable v, deg is an interpretation function, function pt gives the truth degree of $L(v)$ given deg, and cp is the preference function associated with the consequence. The set of all "truth-preference" fuzzy rules will be named $\mathcal{FR}$.

To interpret a conjunction of fuzzy literals, given an interpretation deg, it is natural to take their minimum degree, as usual in conjunctive fuzzy reasoning. Thus function $pt : \mathcal{Q}^* \rightarrow [0, 1]$ will be the min operator.

Definition 2 (pt function)**.** Let $L(v) = \wedge_{i=1,\ldots,n} l_i$, $v \in V$, $l_i \in \mathcal{W} \subseteq \mathcal{Q}$, and $deg : \mathcal{W} \rightarrow [0, 1]$, then $pt(L(v), deg) = min\{deg(l_1), \ldots, deg(l_n)\}$.

For example, a fuzzy proposition A representing sentence "It is hot" can be true with different degrees. We could say it is true with degree $deg(A) = 0.4$ if the outside temperature is mild, and with degree $deg(A) = 0.8$, if the outside temperature is above $80F$. Similarly a fuzzy proposition B representing sentence "I'm thirsty" can reasonably have different truth degrees. We can imagine attaching to a variable v, representing the time at which we go buy a cold drink, label $L(v) = AB$. This will allow us to construct a rule for v which will activate variable "get cold drink" only if the heat level or the thirst are above a given threshold.

Since we will always use the above function pt, each rule can be characterized by its threshold and its preference function. Thus we will sometimes denote a rule via the notation $r(\alpha, cp)$.

Each fuzzy rule states that variable v is part of the problem if value $pt(L(v), deg)$ is greater than the threshold α. Moreover, the consequence specifies the preference associated with the execution of v. In general, such a preference can depend on the truth degree of the premise and on the time at which v is executed. Therefore, it is reasonable to define $cp : [0, 1] \rightarrow (\Re^+ \rightarrow [0, 1])$, that is, as a function which takes in input the truth degree of the premise, i.e., $pt(L(v), deg)$, and returns a function which, in turn, takes in input an execution time and returns a preference in $[0, 1]$.

In other words, function cp allows us to give a preference function on the execution time of v which depends on the truth degree of the label of v. However, this also allows us to model situations where the preference function for the activation of v is independent of the truth degree of the premise, as a special case in which function cp has type $cp : \Re^+ \rightarrow [0, 1]$. This restricted kind of rules will be named r-cp.

In CTPs, a variable without a label implicitly has a label with value true. Similarly, in the fuzzy extension we consider, any variable whose associated rule is not specified has the following implicit one: IF $true$ THEN EXECUTE (v) : 1. This means that variable v is always present in the problem, and its execution has preference 1 independently of the execution time.

Definition 3 (CTPP)**.** A CTPP is a tuple $< V, E, L, R, OV, O, \mathcal{P} >$ where:

- $\mathcal{P}$ is a finite set of fuzzy atomic propositions with truth degrees in $[0, 1]$;
- V is a set of variables;
- E is a set of soft temporal constraints between pairs of variables $v_i \in V$;

- $L : V \rightarrow \mathcal{Q}^*$ is a function attaching conjunctions of fuzzy literals $\mathcal{Q} = \{p_i : p_i \in P\} \cup \{\neg p_i : p_i \in P\}$ to each variable $v_i \in V$;
- $R : V \rightarrow \mathcal{FR}$ is a function attaching a "truth-preference" fuzzy rule $r(\alpha_i, cp)$ to each variable $v_i \in V$;
- $OV \subseteq V$ is the set of observation variables;
- $O : \mathcal{P} \rightarrow OV$ is a bijective function that associates an observation variable to each fuzzy atomic proposition. Variable $O(A)$ provides the truth degree for A.

As explained above, the execution of a variable $v \in V$ depends on the evaluation of the fuzzy rule associated with it. A value assigned to a variable $v \in V$ represents the time at which the action represented by v is executed; this value will be also written as $T(v)$. If v is an observation variable it also represents the time at which the truth degree of the observed proposition is revealed.

Once a CTPP is defined, it is advisable to check statically if the information on labels and rules is consistent similarly to what is done in CTPs. In particular, if a variable v is executed, all the observation variables of the propositions in its label $L(v)$ must have been executed before v. In CTPs this is tested by checking if for each $v \in V$ and for each proposition $A \in L(v)$, $L(v) \supseteq L(O(A))$ and $T(O(A)) < T(v)$, where $O(A)$ is the observation node of proposition A.

In the fuzzy case, where conjunction is replaced by minimum and the truth values of the propositions are in $[0, 1]$, $L(v) \supseteq L(O(A))$ has to be augmented with the condition that the threshold in the rule associated with $O(A)$ should not be lower than the threshold of the rule associated to v. More formally:

Definition 4 (Structural Consistency)**.** Let v be a variable of a CTPP and $L(v)$ its label. A CTPP is **structurally consistent** if each observation variable, say $O(A)$, which evaluates a fuzzy proposition $A \in L(v)$, is such that $L(O(A)) \subseteq L(v)$ and $\alpha \geq \beta$, where $R(v) = r(\alpha, cp)$ and $R(O(A)) = r(\beta, cp')$.

Checking the structural consistency of a CTPP can be performed in $O(|V|^2)$ since to establish the consistency of the label of a variable at most $O(|V|)$ labels (and thresholds) must be considered.

The definitions of scenario, projection, schedule and strategy are analogous to the classical counterparts.

Definition 5 (Scenario)**.** Given an CTPP P with a set of fuzzy literals $\mathcal{Q}$, a **scenario** is an interpretation function $s : \mathcal{W} \rightarrow [0, 1]$ where $\mathcal{W} \subseteq \mathcal{Q}$ that partitions the variables of P in two sets: set V_1, containing the variables that will be executed and set V_2 containing the variables which will not be executed. A variable v, with associated rule $r(\alpha, cp)$, is in V_1 iff $pt(L(v), s) \geq \alpha$, otherwise it is in V_2. $S(P)$ is the set of all scenarios of P.

Definition 6 (Partial scenario)**.** A **partial scenario** is an interpretation function $s : \mathcal{W} \rightarrow [0, 1]$ where $\mathcal{W} \subseteq \mathcal{Q}$ that partitions the variables of the CTPP in three sets: set V_1, containing the variables that will be executed, set V_2 containing the variables which will not be executed and set V_3 containing the variables the execution of which cannot be decided given the information provided by s. A variable v, with associated rule $r(\alpha, cp)$ and label $L(v)$, is in V_3 iff $L(v) \supset \mathcal{W}$, is in V_1 iff $pt(L(v), s) \geq \alpha$, otherwise it is in V_2.

Since a scenario chooses a value for each fuzzy literal, it determines which variables are executed and also which preference function must be used for their execution. This means that a scenario projection must contain the executed variables, the temporal constraints among them, and the information given by the preference function of each of the executed variables. This information can be modelled by additional constraints between the origin of time and the executed variables.

Definition 7 (Constraints induced by a scenario)**.** Given a (possibly partial) scenario s and a variable v executed in s, consider its associated rule $r(\alpha, f) = R(v)$. The constraint induced by this rule in scenario s is the soft temporal constraint $cst_s(v)$ defined on variables x_0 and v by $(0 \leq v - x_0 < +\infty)$ with associated constraint preference function $f(min_{A \in L(v)} s(A))$. The constraints induced by scenario s are all the constraints induced by variables executed in s, that is, $U(s) = \{cst_s(v), v \text{ executed in } s\}$.

Definition 8 (Scenario projection)**.** Given an CTPP P and a scenario (or partial scenario) s of P, its **projection** $Pr(s)$ is the STPP obtained by considering the set of variables of P executed under s, all the constraints among them, and the constraints in $U(s)$. Two scenarios are **equivalent** if they induce the same projection.

Definition 9 (Schedule)**.** A **schedule** $T : V \rightarrow \Re^+$ of a CTPP P is an assignment of execution times to the variables in V. Given a scenario s and a schedule T, the preference degree of T in s is $pref_s(T) = min_{c_{ij} \in Pr(s)} f_{ij}(T(v_j) - T(v_i))$, where f_{ij} is the preference function of constraint c_{ij} defined over variables v_i and v_j. We indicate with $\mathcal{T}$ the set of all schedules.

Given a CTPP P an execution strategy $St : S(P) \rightarrow \mathcal{T}$ is a function from scenarios to schedules.

Figure 2 shows an example of CTPP that extends the CTP in Figure 1. There are three skiing stations: $Sk1$, $Sk2$ and $Sk3$. A represents the fuzzy proposition "there is no snow"; station $Sk1$ is the least accessible, so it is reachable only if A is at least 0.8; on the other hand, station $Sk3$ has the most reliable roads, so it is accessible when A is above 0.3; station $Sk2$ has intermediate reachability conditions, so it is accessible for values of A above 0.5. At the same time, however, the higher the snow, the more preferable it is to go skiing. For this reason, the *cp* functions of the rules are "inversely" proportional to the truth degree of observation A. For example, this function could be $cp(x) = (1 - x)$. The two temporal constraints of the original example from x_0 to $WSk1_e$ and to $WSk3_e$ have been fuzzyfied by using trapezoidal preference functions. The preference functions for the other constraints have been omitted, meaning that they are constant functions always returning 1. In this example there are four distinct scenarios, given by $s_1(A) = 1$, $s_2(A) = 0.8$, $s_3(A) = 0.5$, and $s_4(A) = 0.3$. Thus projection $Pr(s_1)$ is the STPP defined on variables $x_0, HW_s, HW_e, WSk1_s, WSk1_e$, projection $Pr(s_2)$ is

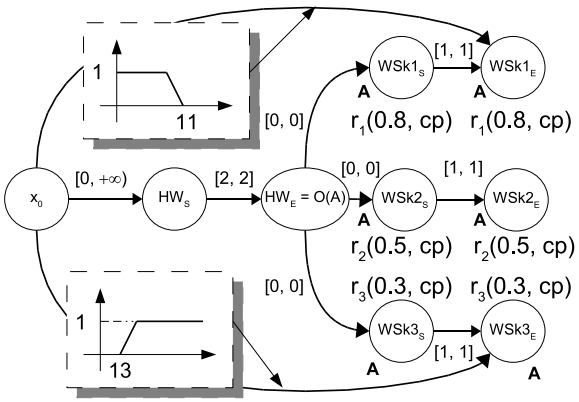

Figure 2: Example of Conditional Temporal Problem with Preferences.

the STPP over variables $x_0, HW_s, HW_e, WSk2_s, WSk2_e$, projection $Pr(s_3)$ is the CTPP over Sx_0, HW_s, HW_e, $WSk3_s, WSk3_e$, and projection $Pr(s_4)$ is the STPP over x_0, HW_s, HW_e.

Consistency notions in CTPPs

Consistency notions in CTPPs are analogous to the ones in CTPs. However, we now have to consider also the preferences. There are again three notions of consistency depending on the assumptions made about the availability of the uncertain information.

Definition 10 (α-Strong Consistency)**.** A CTPP is α-strongly consistent if there is a viable execution strategy St such that, for every scenarios s_1 and s_2, and variable v executed in both,

1. $[St(s_1)](v) = [St(s_2)](v)$;
2. the global preference of $St(s_1)$ and of $St(s_2)$ is at least α.

In words, to be α-strong consistent, we must have a schedule that satisfies all the constraints independently of the observations, giving a global preference greater than or equal to α. This is the strongest consistency notion since it requires the existence of a single schedule that gives preference at least α in every scenario. On the contrary, we can just require the existence for every scenario of a schedule (possibly a different one for different scenarios) that has a preference of at least α given the corresponding projection. This notion is that of α-weak consistency.

Definition 11 (α-Weak Consistency)**.** A CTPP Q is said **α-weakly consistent** (α-WC) if, for every scenario $s \in S(Q)$, $Pr(s)$ is consistent in the STPP sense with preference degree at least α.

The above definitions are at the two extremes w.r.t. assumptions made on which events will be executed: α-SC assumes no knowledge at all, while α-WC assumes the scenario is given. A notion consistency which lies in between is α-dynamic consistency which assumes that the information on which variables are executed becomes available during execution in an on-line fashion. In order to define it, we first need to recall the concept of observation history from CTPs and say when a partial scenario and a scenario are consistent.

Definition 12 (Observation History)**.** Given a scenario s and a schedule T, for each variable v we define the observation history of v w.r.t schedule T and scenario s as the set $H(v, s, T)$ containing the observations performed before time $T(v)$.

Definition 13 (Cons(s,w))**.** Given a CTPP P and scenario s we say a partial scenario w is consistent with s, written $Con(s, w)$ if: STPP $Pr(w)$ is a sub-problem of STPP $Pr(s)$, in the sense that the set of variables (resp. constraints) of $Pr(w)$ is a subset of the set of variables (resp. constraints) of $Pr(s)$ and no variable executed given s is not executed given w.

This last definition extends the one given in the classical case, where it is sufficient to say that a partial assignment is consistent with a scenario if the variables executed by the partial assignment are a subset of those executed by the scenario. We will use this notion in the definition of α-Dynamic Consistency, to express when at a given time the set of observations collected at that time is consistent with a scenario.

Definition 14 (α-Dynamic Consistency)**.** A CTPP is said α-dynamically consistent if there exists a viable execution strategy St such that $\forall v$ and for each pair of scenarios s_1 and s_2 $[Con(s_2, H(v, s_1, St(s_1))) \vee (Con(s_1, H(v, s_2, St(s_2))))] \Rightarrow [St(s_1)](v) = [St(s_2)](v)$ and the global preferences of $St(s_1)$ and $St(s_2)$ are at least α.

In words, a CTPP is α-DC if for every variable v, whenever two scenarios (s_1 and s_2) are not distinguishable at the execution time for v ($Con(s_2, H(v, s_1, St(s_1))) \vee (Con(s_1, H(v, s_2, St(s_2))$), there is an assignment to v ($[St(s_1)](v) = [St(s_2)](v)]$) which can be extended to a complete assignment which in both scenarios will have preference at least α.

It is easy to see that, as for CTPs, α-SC $\Rightarrow$ α-DC $\Rightarrow$ α-WC. Moreover, given $\alpha \in [0, 1]$, if an CTPP is α-SC/DC/WC then it is β-SC/DC/WC $\forall \beta \leq \alpha$.

In what follows we consider a property which is common to all three the consistency notions. In order to do so we consider a subclass of CTPPs characterized by a special type of truth-preference rules. We will then show that the consistency of general CTPPs is equivalent to the consistency of a related problem in such a subclass.

CTPPs with restricted rules. We start by considering a simplified case, that is, when the preference functions of the rules are independent of the truth degree of the label $pt(L(v), deg)$. In such a case, given rule $r(\alpha, f)$, we assume that f is an r-*cp* function. CTPPs with such a restriction will be denoted by R-CTPPs.

The preference information given by f can be equivalently expressed by adding a constraint between the origin of time x_0 and the variable to which rule r is associated. More precisely, the constraint induced by v is the soft temporal constraint $cst(v)$ defined on variables x_0 and v by $(0 \leq v - x_0 < +\infty)$ with associated preference function

$min_{\alpha\in[0,1]} f(\alpha)$. The constraints induced by a whole CTPP Q are all the constraints induced by the variables of Q, that is, $U(Q) = \{cst(v), v \text{ variable of } Q\}$.

In the specific case of an R-CTPP Q, the preference function of each constraint in $U(Q)$ will just be $f(\alpha)$, since in this case f does not depend on the truth value of the propositions in the premise of the rule.

Theorem 1. *Given a CTPP $Q =< V, E, L, R, OV, O, \mathcal{P} >$, let us define a function R' from R as follows: if $R(v) = r(\alpha, f)$, then $R'(v) = r(\alpha, f')$ where $f' = min_{\beta\in[0,1]} f(\beta)$. Then $Q' =< V, E, L, R', OV, O, \mathcal{P} >$ is an R-CTPP. Moreover, Q is α-SC/DC/WC if and only if Q' is α-SC/DC/WC.*

Testing consistency of CTPPs

Thanks to Theorem 1, when testing the consistency of a CTPP we can restrict ourselves to testing the consistency of its related R-CTPP without loss of generality.

Testing α-SC

The algorithm we propose to test the α-SC of an R-CTPP is based on the correspondence of the α-SC of the R-CTPP and the consistency preference degree of a related STPP.

Theorem 2. *Given an R-CTPP $M =< V, E, L, R, OV, O, \mathcal{P} >$, let $E' = E \cup U(M)$. Then M is α-strongly consistent if and only if the STPP $< V, E' >$ is consistent with preference degree α.*

Theorem 2 relates the α-SC of an R-CTPP to the consistency level of an STPP. This allows us to check the α-SC of an R-CTPP by just constructing the appropriate STPP and then finding its best level of consistency. This will give us the highest level α at which the R-CTPP is α-SC. Since, under some tractability assumptions, solving a fuzzy STPP can be done in polynomial time (Khatib *et al.* 2001), $U(Q)$ contains $O(|V|)$ constraints, the procedure takes polynomial time.

Testing α-Weak Consistency

In classical CTPs, the problem of checking WC is a co-$\mathcal{NP}$ complete problem (Tsamardinos, Vidal, & Pollack 2003). Therefore, being CTPPs an extension of CTPs, we cannot expect to do better. The classical algorithm to test the WC of CTPs checks the consistency of all complete scenarios by identifying a set of labels LS that covers all the scenarios (Tsamardinos 2001). As seen in the example in Figure 2, the scenarios of a CTPP are determined not only by the labels used in the problem, but also by the thresholds levels. However, in the case of R-CTPPs, the definition of equivalence between scenarios collapses to that for CTPs, that is, two scenarios are equivalent iff they induce the same partition of the variables. In fact, in R-CTPPs the preference on the induced constraint is independent of the value of the observation in the head of the corresponding rule. Thus the projection of the scenario is fully specified by the set of executed variables.

We first define for each literal $l \in \mathcal{Q}$ an auxiliary set $M(l)$ that contains the set of the threshold levels of truth-preference rules defined on labels containing l. More precisely: $M(l) = \{\alpha_i : \exists v \in V \text{ with } R(v) = r(\alpha_i, cp) \wedge l \in L(v)\} \cup \{1\}$.

Given set $M(l)$ for each literal l, we consider scenarios mapping each literal l into a value in $M(l)$.

Definition 15 (Meta-scenario)**.** Given a CTPP P with set of fuzzy literals $\mathcal{Q}$ a meta-scenario is an interpretation function $ms : (\mathcal{W} \subseteq \mathcal{Q}) \rightarrow \cup_{l\in\mathcal{W}} M(l)$ such that $ms(l) \in M(l)$, $\forall l \in \mathcal{W}$. We will denote the set of meta-scenarios as $MS(P) \subset S(P)$.

Given the equivalence relation defined on R-CTPP scenarios, every scenario $s \in S(P) \setminus MS(P)$ is equivalent to a meta-scenario $ms \in MS(P)$.

Theorem 3. *Given an R-CTPP P, $\forall s \in S(P)$, $\exists ms \in MS(P)$ s.t. $Pr(s) = Pr(ms)$.*

In particular, from the above theorem we can immediately deduce that a R-CTPP is α-WC if and only if all projections of meta-scenarios are consistent with optimal preference level at least α. However, two meta-scenarios in $MS(P)$ can be equivalent. In order to further reduce the set of projections to be considered, we apply a procedure similar to that proposed in (Tsamardinos, Vidal, & Pollack 2003), in order to find a minimal set of meta-scenarios containing only one meta-scenario for each equivalence class. We refere to this procedure as Algorithm *FST*.

Algorithm *FST* takes in input a set of propositions $\mathcal{SL}$, a current partial meta-scenario s, the set $ExecVars$ of variables which can be executed given the information in s, the set PV containing the sets of executed variables already considered, and, finally, the set MS of meta-scenarios selected so far. In output, it gives set of meta-scenarios MS'. First considers if the set of propositions $\mathcal{SL}$ is empty and, if so, it returns the current set of meta-scenarios MS. Otherwise, it chooses (in some pre-fixed order) proposition H and then removes it from $\mathcal{SL}$. Next, for each threshold α (in increasing order) in the set $M(H)$, it extends the current meta-scenario with assignment $H = \alpha$ and computes the set of variables $ExecVars$ which are or could be executed given the information in s. In more detail, procedure $ConsVars$ takes in input a set of variables X, a partial meta-scenario w, and a CTPP P, and returns the subset of variables of X containing only variables that in P are associated with a rule whose head is not false given w (set $V_1 \cup V_3$ according to the notation of Definition 6).

If set $ExecVars$ has not been considered before (that is, it is not contained in set PV) then, if either all the propositions in $\mathcal{SL}$ have been considered or $ExecVars$ is empty, then $ExecVars$ is added to set PV and the set of meta-scenarios MS is updated with the new meta-scenario found s. Otherwise, if neither of the above sets are empty the search is carried on recursively.

In order to find a minimal set of meta-scenarios of an R-CTPP P with proposition set $\mathcal{P}$, Algorithm *FST* is called with $\mathcal{SL} = \mathcal{P}$, $s = nil$[1] $ExecVars = V$, $PV = \emptyset$, $MS = \emptyset$.

[1]We write $s = nil$ meaning the function with the empty domain, that is, to model a partial scenario in which no proposition is assigned.

The key idea of the algorithm is that as we extend a partial scenario the set of variables that could be executed can only shrink. Moreover, since for each proposition H the thresholds in $M(H)$ are considered in increasing order, when a set of executed variables is found, all its subsets have already been considered and thus if such a set is already in PV the search can avoid the recursive call.

Theorem 4. *Consider an R-CTPP P with proposition set $\mathcal{P}$. Let MS' be the set of meta-scenarios returned by Algorithm* FST *when called on $\mathcal{SL} = \mathcal{P}$, $s = nil$, $ExecVars = V$, $PV = \emptyset$, $MS = \emptyset$. Then:*

- $\forall s \in MS'$, $s \in MS(P)$;
- $\forall s' \in MS(P)$, $\exists s \in MS'$ *such that* $Pr(s') = Pr(s)$;
- $\forall s, s' \in MS'$, $Pr(s) \neq Pr(s')$;

The complexity of Algorithm *FST* is $O(\Pi_{H\in\ mathcalSL}|M(H)|)$ since, in the worst case the algorithm explores the whole set of meta-scenarios, of size $\Pi_{H\in\mathcal{SL}}|M(H)|$.

example Consider the following R-CTPP with four variables v_1, v_2, v_3, v_4 whose associated rules are $R(v_1) = r(0.3, \text{IF } A > 0.3 \text{ THEN EXECUTE } v_1 : 1)$, $R(v_2) = r(0.5, \text{IF } A > 0.5 \text{ THEN EXECUTE } v_2 : 1)$, $R(v_3) = r(0.2, \text{IF } AB > 0.2 \text{ THEN EXECUTE } v_3 : 1)$, and $R(v_4) = r(0.5, \text{IF } AB > 0.5 \text{ THEN EXECUTE } v_4 : 1)$. In this case $M(A) = \{0.2, 0.3, 0.5, 1\}$ and $M(B) = \{0.2, 0.5, 1\}$. This problem has 12 meta-scenarios, while the minimal set is $\{\{A = 0.2\}, \{A = 0.3, B = 0.2\}, \{A = 0.5, B = 0.2\}, \{A = 0.5, B = 0.5\}, \{A = 1, B = 0.2\}, \{A = 1, B = 0.5\}, \{A = 1, B = 1\}\}$.

The algorithm we propose to test α-WC of a R-CTPPcomputes a minimal set of meta-scenarios applying Algorithm *FST* and for each such meta-scenario ms it checks if the corresponding projection $Pr(ms)$ is consistent at level α. If the preference functions are semi-convex, in order to test this it is sufficient to test whether the STP obtained from $Pr(s)$ via its α-cut (that is considering for each constraint the sub-interval containing elements mapped into a preference $\geq \alpha$) is consistent.

If the preference functions are semi-convex the co-problem of α-WC is NP-complete since it coincides with deciding if there is an inconsistent STP obtained via the α-cuts. Thus in such a case testing α-WC is co-NP-complete.

Testing α-Dynamic Consistency

In (Tsamardinos, Vidal, & Pollack 2003) the DC of a CTP is checked by transforming the CTP into a Disjoint Temporal Problem (DTP) (Stergiou & Koubarakis 2000) obtained from the union of the STPs corresponding to the projections of the scenarios of the CTP and some additional disjunctive constraints. A CTP is DC if, whenever at certain point in time a given variable must be executed, and it is not possible to distinguish in which scenario we are, there is a value to assign to such a variable which will be consistent with all the possible scenarios that can evolve in future. This means that all the variables representing the same CTP variable in the projections either are constrained to be after observations which allow to distinguish the scenario univocally (and thus can be executed independently of each other) or they must be assigned the same value whenever observation variables do not allow to distinguish the scenarios. This is modeled by adding to the STP, obtained by the union of all the projections of the CTP, a specific set containing disjunctive constraints (called DC constraints). Briefly, each DC-constraint regarding a variable v requires that in all scenarios either the execution of v follows that of all the observation variables of literals in its label, $L(v)$, or, otherwise, that the occurrences of v are synchronized.

Adding DC constraints makes the STP become a DTP (see (Tsamardinos, Vidal, & Pollack 2003) for more details).

Since in R-CTPPs executing a variable at the same time in different scenarios gives the same preference, the reasoning above can be applied directly. In fact, in terms of synchronization only the temporal order matters.

Theorem 5. *Given an R-CTPP $Q =< V, E, L, R, OV, O, \mathcal{P} >$, let $D = \langle V', E' \rangle$ be the fuzzy DTP with $V' = (\bigcup_{Pr(s)=(V,E), s\in MS'} V)$ and $E' = (\bigcup_{Pr(s)=(V,E), s\in MS'} E) \cup CD$. Then Q is α-dynamically consistent if and only if D is consistent with preference degree α.*

Theorem 5 allows us to define an algorithm which, given in input an R-CTPP, tests if it is α-DC. Such an algorithm first computes the minimal set of meta-scenarios by applying Algorithm *FST*. Next, it tests if the DTPP obtained taking the union of the all the STPPs corresponding to projections of meta-scenarios in the minimal set, and adding the CD constraints, is consistent with optimal preference level α. Thus the complexity of checking α-DC is the same as that of solving a fuzzy DTPP; we recall that efficient algorithms for finding the optimal preference level of Fuzzy DTPPs have been considered in (Peintner & Pollack 2004).

CTPPs vs. STPPUs

It is interesting to notice that consistency in CTPs is strongly connected to controllability in STPUs. This arises from the fact that both kinds of problems are concerned with the representation of uncertainty: STPUs model uncertainty by defining contingent constraints, while CTPs try to capture the outcomes of external events by modelling conditional executions.

We propose here a mapping from STPPUs to CTPPs that preserves the controllability/consistency of the problem. The main idea of this mapping is that, if an STPU has contingent constraints defined over finite domains, each possible value that their endpoints can assume is, in a sense, a condition which has been satisfied.

Given an STPPU $Q =< N_e, N_c, L_r, L_c >$, let $k = |L_c|$, for every soft contingent temporal constraints $l_i \in L_c$ such that $l_i =< [a_i, b_i], f_i >$ we discretize the interval $[a_i, b_i]$ and we denote the number of elements obtained with $|l_i|$ indicating such a set of elements with $\{d_{ij}, j = 1 \ldots |l_i|\}$. For the sake of notation, we write $I = \{1 \ldots |L_c|\}$ and, for each $i \in I$, $J_i = \{1, \ldots, |l_i|\}$

Let us consider the mapping applied to a contingent constraint $l_i =< [a_i, b_i], f_i >$, defined on executable A and contingent variable C. We add $|l_i|$ observation variables, o_{ij}, and $|l_i|$ variables v_{ij}, one for each possible occurrence of C at time d_{ij} in $[a_i, b_i]$. Variable o_{ij} observes the proposition $p_{ij} =$ "$C = d''_{ij}$, while variable v_{ij} represents the actual occurrence of C at time d_{ij}.

Moreover we add a hard temporal constraint with interval $e_{ij} =< [0, 0], 1 >$ between o_{ij} and v_{ij}, and and we add a soft constraint $eo_{ij} =< [d_{ij}, d_{ij}], f_{|d_{ij}} >$ between A and o_{ij}.

Any other constraint w involving C in the STPPU is replicated $|l_i|$ times, one for each d_{ij}, obtaining constraint w_{ij} connected to the corresponding v_{ij} variable.

Definition 16. Given an STPPU $Q =< N_e, N_c, L_r, L_c >$, where I and J_i are as above, we define the CTPP $C(Q)$ as the tuple $< V, E, L, R, OV, O, \mathcal{P} >$, where

- $\mathcal{P}$ is the set of fuzzy atomic propositions $\{p_{ij}, i \in I, j \in J_i\}$;
- $V = N_e \cup \{o_{ij}, i \in I, j \in J_i\} \cup \{v_{ij}, i \in I, j \in J_i\}$;
- $E = L^e_r \cup \{e_{ij}, i \in I, j \in J_i\} \cup \{eo_{ij}, i \in I, j \in J_i\} \cup \{w_{ij}, i \in I, j \in J_i\}$ where L^e_r is the set of all the requirement constraints in L_r defined only between executable variables and e_{ij}, eo_{ij}, and w_{ij} are as defined above;
- $L : V \rightarrow \mathcal{Q}^*$ is a function such that $L(v_{ij}) = p_{ij}$ and $true$ otherwise;
- $R : V \rightarrow \mathcal{FR}$ is a function defined as $R(v_{ij}) = r(0, g)$, where g is the constant function equal to $f(d_{ij})$ where is the preference function of l_i;
- $OV \subseteq V$ is the set of observation variables $\{o_{ij} \in I, j \in J_i\}$;
- $O : \mathcal{P} \rightarrow OV$ is a bijective function such that $O(p_{ij}) = o_{ij}$;

It is possible to show that this mapping preserves the controllability/consistency notions.

Theorem 6. *Given an STPPU Q and its corresponding CTPP $C(Q)$, Q is α-strongly (resp., weakly, dynamically) controllable iff $C(Q)$ is α-strongly (resp., weakly, dynamically) consistent.*

Notice that the result above mentions α-weak controllability, which is not defined in (Rossi, Venable, & Yorke-Smith 2006), where only the stronger notion of Optimal-weak controllability is considered. However α-weak controllability can be directly obtained from the definition of Optimal weak controllability by replacing "optimal" with "$\geq \alpha$ whenever the projection has optimal preference at least α".

References

Ai-Chang, M.; Bresina, J. L.; Charest, L.; Chase, A.; jung Hsu, J. C.; Jónsson, A. K.; Kanefsky, B.; Morris, P. H.; Rajan, K.; Yglesias, J.; Chafin, B. G.; Dias, W. C.; and Maldagu, P. F. 2004. Mapgen: Mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems* 19(1):8–12.

Combi, C., and Pozzi, G. 2006. Task scheduling for a temporal workflow management system. In *Proc. of TIME'06*, 61–68.

Cox, E. 1992. Fuzzy fundamentals. *IEEE Spectrum* 29(10)(10):58–61.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.

Dubois, D., and Prade, H. 1996. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems* 84:169–185.

Khatib, L.; Morris, P.; Morris, R. A.; and Rossi, F. 2001. Temporal constraint reasoning with preferences. In *IJCAI01*, 322–327.

Lee, C. C. 1990. Fuzzy logic in control systems: fuzzy controllers. *IEEE Trans. on Sys., Man and Cybern.* 2092:404–435.

Peintner, B., and Pollack, M. E. 2004. Low-cost addition of preferences to dtps and tcsps. In *Proc. of AAAI'-04*, 723–728. AAAI Press / The MIT Press.

Pollack, M. E.; Brown, L. E.; Colbry, D.; McCarthy, C. E.; Orosz, C.; Peintner, B.; Ramakrishnan, S.; and Tsamardinos, I. 2003. Autominder: an intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems* 44(3-4):273–282.

Rossi, F.; Venable, K. B.; and Yorke-Smith, N. 2006. Uncertainty in soft temporal constraint problems: a general framework and controllability algorithms for the fuzzy case. *Journal of AI Research* 27:617–674.

Stergiou, K., and Koubarakis, M. 2000. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence* 120(1):81–117.

Tsamardinos, I.; Vidal, T.; and Pollack, M. E. 2003. Ctp: A new constraint-based formalism for conditional, temporal planning. *Constraints* 8(4):365–388.

Tsamardinos, I. 2001. *Constraint-Based Temporal Reasoning Algorithms with Applications to Planning*. Ph.D. Dissertation, University of Pittsburgh.

Vidal, T., and Fargier, H. 1999. Handling contigency in temporal constraint networks. *J. of Experimental and Theoretical Artificial Intelligence Research* 11(1):23–45.