# Temporal Reasoning with Preferences and Uncertainty

**N. Yorke-Smith**
IC–Parc
Imperial College London
London, U.K.
nys@icparc.ic.ac.uk*

**K. B. Venable** and **F. Rossi**
Dept. of Mathematics
University of Padova
Padova, Italy
{kvenable,frossi}@math.unipd.it†

## Abstract

Temporal Constraint Satisfaction Problems allow for reasoning with events happening over time. Their expressiveness has been extended independently in two directions: to account for uncontrollable events, and, more recently, to account for soft temporal preferences. The motivation for both extensions is from real-life temporal problems; and indeed such problems may well necessitate *both* preferences and uncertainty. This paper proposes the study of temporal problems with both preferences and uncertainty, and puts forward some methods for their resolution.

## 1 Motivation and Background

Research on temporal reasoning, once exposed to the difficulties of real-life problems, can be found lacking both expressiveness and flexibility. Planning and scheduling for satellite observations, for instance, involves not only quantitative temporal constraints between events and qualitative temporal ordering of events, but also soft temporal preferences and contingent events over which the agent has no control. For example, on one hand, slewing and scanning activities should not overlap, but may if necessary. On the other hand, the duration of failure recovery procedures is not under the direct control of the satellite executive. To address the lack of expressiveness, preferences can be added to the framework; to address the lack of flexibility to contingency, handling of uncertainty can be added. Some real-world problems, however, have need for both. It is this requirement that motivates us.

In a temporal constraint problem, as defined in Dechter *et al.* [1991], variables denote timepoints and constraints represent the possible temporal relations between them. Such constraints are quantitative, describing restrictions on either durations or distances of events, in terms of intervals over the timeline. In general such problems are NP-complete. However, if each temporal constraint has just one interval — hence the constraints have form $l_{ij} \leq x_j - x_i \leq u_{ij}$, where $x_i$ denote the timepoints — then we have a *Simple Temporal Problem* (STP) that can be solved in polynomial time.

*Partially supported by the EPSRC under grant GR/N64373/01.
†Partially funded by the Italian MIUR projects NAPOLI and COVER, and the ASI (Italian Space Agency) project ARISCOM.

To address the lack of expressiveness in standard STPs, the *Simple Temporal Problems with Preferences* (STPP) framework merge STPs with semiring-based soft constraints [Rossi *et al.*, 2002]. Soft temporal constraints are specified by means of a *preference function* on the constraint interval, $f : [l_{ij}, u_{ij}] \rightarrow A$, where $A$ is a set of preference values. The set $A$ is part of a semiring. In general, STPPs are NP-complete. However, if the preference functions are semi-convex (that is, they have at most one peak), the constraints are combined via an idempotent operation (like max or min), and the preference values are totally ordered, then finding an optimal solution of a STPP is a polynomial problem. Two solvers for STPPs with the features above are presented by Rossi *et al.* [2002]. The first, Path-solver, enforces path consistency in the constraint network, then takes the subinterval on each constraint corresponding to the best preference level. This gives a standard STP, which is then solved for the first solution by backtrack-free search. The second solver, Chop-solver, is less general but more efficient. It finds the maximum level $y$ at which the preference functions can be 'chopped', i.e. reduced to the set $\{x : x \in I, f(x) \geq y\}$. This set is a simple interval for each $I$. Hence we obtain a standard STP, $STP_y$. By binary search, the solver finds the maximal $y$ for which $STP_y$ is consistent. The solutions of this STP are the optimal solutions of the original STPP.

To address the lack of flexibility in execution of standard STPs, Vidal and Fargier [1999] introduced *Simple Temporal Problems under Uncertainty* (STPUs). Here, again as in a STP, the activities have durations given by intervals. The timepoints, however, fall into two classes: requirement and contingent. The former, as in a STP, are decided by the agent, but the latter are decided by 'nature': the agent has no control over when the activity will end; he observes rather than executes. The only information known prior to observation is that nature will respect the interval on the duration. *Controllability* of a STPU is the analogue of consistency of a STP. Controllable implies the agent has a means to execute the timepoints under his control, subject to all constraints. Three notions are proposed. A STPU is *strongly controllable* if there is a fixed execution strategy that works in all realisations (that is, an observation of all contingent timepoints). Checking strong controllability is in P [Vidal and Fargier, 1999]. A STPU is *dynamically controllable* if there is a online execution strategy that depends only on observed timepoints in the

past and that can always be extended to a complete schedule whatever may happen in the future. Checking dynamic controllability is also in P [Morris *et al.*, 2001]. A STPU is *weakly controllable* if there exists at least one execution strategy for every realisation. Checking weak controllability is co-NP [Vidal and Fargier, 1999]. The three notions are ordered by their strength: strong $\Rightarrow$ dynamic $\Rightarrow$ weak.

In this paper we introduce a framework which allows one to handle both preferences and uncertainty in temporal problems. This is done by just merging the two pre-existing models of STPPs and STPUs. We also adopt the notion of controllability, to be used instead of consistency because of the presence of uncertainty, and we adapt it to handle preferences. We then investigate the complexity of checking such notions of controllability, and we develop algorithms to perform such checks. The main results are for strong and weak controllability, which we show to maintain the same complexity as in STPUs under the same hypothesis that make STPPs polynomially solvable. As future work, we plan to study dynamic controllability, and to implement our algorithms to compare their performance experimentally.

## 2 Simple Temporal Problems with Preferences and Uncertainty (STPPU)

A STPPU is a STPP with timepoints partitioned into two classes, requirement and contingent, just as in a STPU. Since some timepoints are not controllable by the agent, the notion of consistency of a STP(P) is replaced by controllability, just as in a STPU, with the three notions suitably extended. While all three notions of controllability used for STPUs can be adapted to STPPUs, for this short paper we will focus on just weak and strong controllability. A STPPU is *optimally strongly controllable* if there is a fixed execution strategy that works in all realisations and is optimal in each of them. Rather than optimality, one could be interested in just reaching a certain quality level: a STPPU is *$\alpha$-strongly controllable*, where $\alpha$ is a preference level, if there is a fixed execution strategy that works in all realisations and is above $\alpha$ in each of them. A STPPU is *optimally (resp. $\alpha$-) weakly controllable* if there exists at least one optimal (resp., above $\alpha$) execution strategy for every realisation.

## 3 Checking Optimal and $\alpha$ Strong Controllability

It can be proved that checking either optimal or $\alpha$ strong controllability is in P. The method we propose relies on two known algorithms. The first is Path-solver, which enforces path consistency on a STPP. The second is Strong-Controllability(STPU) [Vidal and Fargier, 1999], which checks if a STPU is strongly controllable. The main idea is to apply Strong-Controllability(STPU) to a special STPU, which we will call $P_{opt}$, that can be constructed starting from the STPP $P'$ obtained by applying Path-solver to a STPPU $P$. More precisely, the algorithm takes as input a STPPU $P$ and performs the following steps: (1) forgetting the uncertainty, treat $P$ as a STPP and apply Path-solver to it, obtaining the STPP $P'$; (2) collapse the non-contingent intervals of $P'$ to those parts which have the highest preference level, and then neglect the preferences, obtaining a STPU $P_{opt}$; (3) apply Strong-Controllability to $P_{opt}$. Theoretical results that we have proven, but omit for lack of space, show the correctness of the algorithm: (1) If a STPPU is optimally strong controllable, then the STPU obtained by neglecting preferences is strongly controllable; (2) A STPPU $P$ is optimally strong controllable if and only if the STPU $P_{opt}$ is strongly controllable.

We have also generated a different algorithm by combining Strong-Controllability with Chop-solver. This algorithm is less general but more efficient. Both algorithms we propose are polynomial, with a time complexity $O(n^3 \times R \times l)$, where $n$ is the problem size, $R$ is the range of the largest interval, and $l$ is the number of preference levels.

For $\alpha$-strong controllability, we rely on theoretical results similar to those above, and we propose two algorithms. Given $\alpha$, the first one checks $\alpha$-SC by: cutting the intervals to those parts whose preference is above $\alpha$, and applying strong controllability checking to the resulting STPU. The time complexity is $O(n^3 \times R)$. The second algorithm finds the highest $\alpha$ at which $P$ is $\alpha$-SC, by performing a binary search for the highest $\alpha$, checking strong controllability at each step. Its time complexity is $O(p \times n^3 \times R)$, where $p$ is proportional to the precision we want in returning the result.

## 4 Checking Optimal and $\alpha$ Weak Controllability

Optimal weak controllability of a STPPU is equivalent to weak controllability of the corresponding preference-stripped STPU. Thus we can use the existing STPU algorithms for checking weak controllability. For checking $\alpha$-weak controllability, we propose two approaches. The first chops the STPPU at level $\alpha$ by considering only those parts of its intervals which have preference level above $\alpha$ (semi-convexivity guarantees that the result is always a single interval), and then applies Weak-Controllability to the STPU so obtained. The second approach generates the realisations that arise by considering the contingent intervals reduced to just their upper or lower bound (other realisations need not be considered) [Vidal and Fargier, 1999]), and checks the consistency of all such realisations. Both algorithms are exponential in the number of contingent constraints.

## References


[Dechter *et al.*, 1991] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[Morris *et al.*, 2001] Paul Morris, Nicola Muscettola, and Thierry Vidal. Dynamic control of plans with temporal uncertainty. In *Proc. of IJCAI'01*, pages 494–502, Seattle, WA, 2001.

[Rossi *et al.*, 2002] F. Rossi, A. Sperduti, K.B. Venable, L. Khatib, P. Morris, and R. Morris. Learning and solving soft temporal constraints: An experimental study. In *Proc. of CP'02*, pages 249–263, Ithaca, NY, 2002.

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