Temporal Constraints and Uncertainty

Ioannis Tsamardinos

Department of Biomedical Informatics, Vanderbilt University
tsamard@eecs.umich.edu

Abstract. We present a summary of the state of the art in Constrained-Based Temporal Reasoning focusing on the treatment of Uncertainty. Current trends, on-going work, open problems and suggestions for future work are also presented. Applications of this kind of research span areas such as planning, scheduling, workflow management systems, real-time systems, and temporal databases.

One of the most important formalization in Constraint-Based Temporal reasoning is the Simple Temporal Problem (STP) (see chapter 3 in [1] for an introduction) in which there is a set of time-points $x, y, \ldots$ and constraints among them of the form $lb \leq x - y \leq ub$. The question then is to find whether there is a satisfying assignment to the time-points (**consistency checking**). The STP's popularity stems from the fact that consistency checking is polynomial. Nevertheless, the STP is quite restrictive because it does not allow the expression of disjunctions. Reasoning with disjunctions and n-ary constraints is essential to many applications, such as Planners and Schedulers, to express for example, the fact that two tasks should not overlap. STPs were later extended to include n-ary disjunctive constraints in a new formalism called the Disjunctive Temporal Problem (DTP). However, consistency checking in the later is NP-complete; nevertheless, recent work on DTP solving has significantly improved performance [1].

Algorithms for the STP and DTP answer the question whether there is a satisfying assignment, but this is useful only if it is the agent that selects the times to assign. In real applications many events are **uncontrollable** and their exact timing uncertain, i.e. it is Nature that selects the time to assign. Planners and Schedulers have to execute STPs or DTPs so that whatever Nature selects for the future uncontrollables, they can keep executing respecting the constraints. The problem then becomes finding an **execution strategy** (policy) mapping from the current assigned times (history) to the next assignment to make (controllability). Another type of uncertainty deals with the outcome of observations and **conditional execution**. In conditional plans actions are executed depending on the outcome of certain observations. STPs or DTPs do not allow the specification of conditional events and are inadequate as conditional models. Again the problem is to discover an execution policy that no matter what the outcome of the observations turn out to be, the planner can keep executing respecting the constraints.

The problem with uncontrollables was first attacked by planners with a "push under the carpet" approach. Instead of finding and executing a predetermined exact solution schedule of an STP, bounds on the allowable execution times of the time-points were maintained and the STP was dynamically solved during execution. As long as Nature was selecting times within the bounds, execution was successful. Later this approach was extended for executing DTPs which can maintain greater flexibility on the bounds [1].

The first model to explicitly deal with uncontrollables was the STPU (STP with Uncertainty). Surprisingly, finding an execution strategy turned out to be a polynomial problem [2]. The first model to deal with conditional events was the CSTP (Conditional STP) and its disjunctive counter-part CDTP [1]. A polynomial algorithm for the CSTP exists only for problems that follow certain structural requirements, but fortunately it is expected that future conditional temporal planners will produce such CSTPs. Therefore, we expect that in practice most CSTPs can be solved in polynomial time.

A **hybrid** model that encodes and reasons with uncontrollable and conditional events and disjunctive constraints would consist a significant step forward in temporal reasoning with uncertainty. Even with such a hybrid though, there are still many open questions regarding the handling of uncontrollable events. Each STPU constraint that involves an uncontrollable event x is of the form $lb \leq x - y \leq ub$, e.g. if x is the end of the event and y its beginning the constraint specifies that the duration will be in the interval $[lb, ub]$. But how do we derive the bounds lb and ub? The short answer is domain knowledge. Inherently however, *there are no such bounds*: a task always has a small probability of taking longer than ub and shorter than lb. The domain encoder has the difficult task of selecting these bounds. If they are too tight, the probability of Nature not respecting them (and the execution strategy of the STPU becoming irrelevant) is significant; if they are too loose, the probability of the resulting STPU to be controllable is reduced. Notice that the STPU algorithms will only determine if there is an execution strategy or not given the bounds: if there is none we get no information as to how to actually execute the STPU.

This is an important issue for future research. One approach would be a decision theoretic one in which the constraint $lb \leq x - y \leq ub$ is sustituted with a probability distribution on the time of occurence of x given the time the agent assigned to the controllable y. Another approach would be to develop algorithms that instead of returning a yes/no answer to the controllability property of an STPU, determine the "best" bounds for which the STPU is controllable. We are currently investigating both of these approaches.

References

1. Ioannis Tsamardinos. *Constraint-based Temporal Reasoning Algorithms with Applications to Planning.* PhD thesis, University of Pittsburgh, August 2001.
2. Thierry Vidal and Paul Morris. Dynamic control of plans with temporal uncertainty. In *International Joint Conference on Artificial Intelligence, IJCAI-2001*, 2001.