Simple Temporal Problems with Preferences and Uncertainty

Kristen Brent Venable
Dept. of Pure and Applied Mathematics
University of Padova, Italy

Neil Yorke-Smith
IC–Parc
Imperial College London, U.K.

Abstract

Simple Temporal Problems (STPs) are a restriction of the framework of Temporal Constraint Satisfaction Problems, tractable in polynomial time. Their expressiveness has been extended independently in two ways. First, to account for uncontrollable events, to Simple Temporal Problems with Uncertainty (STPUs). Second, more recently, to account for soft temporal preferences, to Simple Temporal Problems with Preferences (STPPs). The motivation for both extensions is from real-life problems; and indeed such problems may well necessitate *both* preferences and uncertainty. Our research proposes the study of Simple Temporal Problems with Preferences and Uncertainty (STPPUs), and puts forward two notions of controllability for their resolution.

Motivation

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 example, 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, 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 of hard constraints, 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 here.

Background

Temporal Constraint Satisfaction Problems

In a temporal constraint problem, variables denote timepoints or intervals, and constraints represent the possible temporal relations between them. Temporal constraints can be quantitative (distance between timepoints) or qualitative (relative position of temporal objects).

Dechter, Meiri, & Pearl (1991) introduced the quantitative *Temporal CSP* (TCSP) and its restricted subclass, the *Simple Temporal Problem* (STP). Variables x_i represent timepoints (events) and constraints represent the relations between them. The restriction to at most one interval in each temporal constraint — hence the constraints have form $l_{ij} \leq x_j - x_i \leq u_{ij}$ — entails that a STP can be solved in polynomial time. By solved, we mean that consistency is decided and the minimal network obtained; applying path consistency suffices for this. In contrast, the general TCSP is NP-complete.

Simple Temporal Problems with Preferences

To address the lack of expressiveness in standard STPs, Khatib *et al.* (2001) introduced the *Simple Temporal Problem with Preferences* (STPP). The framework merges temporal CSP with the standard semiring-based soft constraints (Bistarelli, Montanari, & Rossi 1997). In addition to the hard temporal constraints $l_{ij} \leq x_j - x_i \leq u_{ij}$ of a STP, soft temporal constraints are specified by means of a *preference function* on an interval, $f : I \to A$, where $I = [l_{ij}, u_{ij}]$ and A is a set of preference values, part of a semiring $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$. (A semiring is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ such that: A is a set and $\mathbf{0}, \mathbf{1} \in A$; $+$ is commutative, associative and $\mathbf{0}$ is its unit element; $\times$ is associative, distributes over $+$, $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element. A *c-semiring* is a semiring in which $+$ is idempotent, $\mathbf{1}$ is its absorbing element and $\times$ is commutative.)

In general, solving a STPP is NP-complete. However, on making three assumptions, notably one about the shape of the preference functions, solving is polynomial in the number of timepoints: (1) the preference functions are semi-convex, (2) the semiring multiplicative operator is idempotent, and (3) the values of the semiring are totally ordered.

Rossi *et al.* (2002b) present two solvers for STPP. The first, Path-solver, enforces path consistency in the constraint network, then takes the sub-interval on each constraint corresponding to the best preference level. This gives a standard STP, which is then solved for the first solution by back-free search. The complexity is polynomial, but the performance can be poor because a pointwise (discrete) representation is used for the intervals and the preference functions. The second solver, Chop-Solver, is less general but more efficient. It finds the maximum level α at which the preferences can be 'chopped', i.e. the intervals are reduced to the set $\{x : x \in I, f(x) \geq \alpha\}$ of values mapped to at least α by the preference functions. This set is a simple interval for each

I, provided the three above assumptions hold. Hence we obtain a standard STP, STP_α. By binary search, the solver finds the maximal α for which STP_α is consistent. The solutions of this STP are the solutions of the original STPP.

Simple Temporal Problems under Uncertainty

To address the lack of flexibility in execution of standard STPs, Vidal & Fargier (1999) introduced the *Simple Temporal Problem under Uncertainty* (STPU).

Here, as in a STP, the activities have durations specified by intervals. The start times of all activities are assumed controlled by the agent (this brings no loss of generality). The end times, 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 task will end; he observes rather than executes. The only information known prior to observation is that nature will respect the interval on the duration. Durations of contingent links are assumed independent.

Consistency is not enough to ensure temporal feasibility in the presence of contingent events. Rather, the stronger notion of *controllability* of a STPU is the analogue of consistency of a STP. Controllable means the agent has a strategy to execute the timepoints under his control, subject to all constraints, in all situations involving the contingent timepoints. Three levels of controllability are proposed:

- A STPU is *strongly controllable* if there is a fixed execution strategy that works in all realisations. (A *realisation* is a possible outcome of the world, i.e. in this case, an observation of all contingent timepoints.)
- 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.
- A STPU is *weakly controllable* if there is a viable global execution strategy: there exists at least one schedule for every realisation.

The three notions are ordered by their strength: strong $\Rightarrow$ dynamic $\Rightarrow$ weak. The first requires no knowledge of the realisation, and is in P. The second, surprisingly, is also in P (Morris, Muscettola, & Vidal 2001). It is seen as the most realistic knowledge assumption in many practical cases, since it interleaves scheduling, observation and execution. The third requires a prior knowledge of the realisation, and is co-NP complete.

In this paper we formally define a class of temporal constraint satisfaction problems that feature both preferences and uncertainty. For this class of problems we consider the equivalent of Strong and Weak Controllability. In particular we extend both notions of controllability and we give algorithms to check whether a problem satisfies their definition. We show that adding preferences does not impact on the complexity of checking these two types of controllability. In fact, the algorithms we propose for checking strong controllability of STPPUs with preferences are polynomial, while those for checking weak controllability are co-NP complete.

Simple Temporal Problems with Preferences and Uncertainty

Consider a temporal problem that we would model naturally with preferences, in addition to hard constraints, but that also features uncertainty. Neither a STPP nor a STPU is adequate. Therefore we propose what we call *Simple Temporal Problems with Preferences and Uncertainty*, or STPPUs.

An informal definition of a STPPU is a STPP for which the (end) timepoints are 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. Every solution to the STPPU has a global preference value, just as in a STPP, and we seek a solution which maximises this value.

More precisely, we can extend some definitions given for STPPs and STPUs to fit STPPUs in the following way:

Definition 1 • executable timepoints *are those points, b_i, whose date is assigned by the agent;*

- contingent timepoints *are those points, e_i, whose uncontrollable date is assigned by the external world;*
- generic timepoint t_i *is either an executable or a contingent timepoint;*
- decision $\delta(b_i)$ *is a value assigned to an executable timepoint;*
- observation $\omega(e_i)$ *is a value assigned (by Nature) to a contingent timepoint;*
- assignment $\gamma(t_i)$ *is a value assigned by either a decision to an executable timepoint or by an observation to a contingent timepoint;*
- *a* soft requirement constraint r_{ij}, *on generic timepoints t_i and t_j, is a pair $\langle I_{ij}, f_{ij}\rangle$, where $I_{ij} = [l_{ij}, u_{ij}]$ such that $l_{ij} \leq \gamma(t_j) - \gamma(t_i) \leq u_{ij}$, and $f_{ij} : I_{ij} \rightarrow A$ is a* requirement preference function *mapping each element of the interval into an element of the preference set of the semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$;*
- *a* soft contingent constraint g_{ij}, *on executable point b_i and contingent point e_j, is a pair $\langle \hat{I}_{ij}, \hat{f}_{ij}\rangle$ where $\hat{I}_{ij} = [\hat{l}_{ij}, \hat{u}_{ij}]$ such that $\hat{l}_{ij} \leq \omega(e_j) - \delta(b_i) \leq \hat{u}_{ij}$ and $\hat{f}_{ij} : \hat{I}_{ij} \rightarrow A$ is a* contingent preference function *that maps each element of the interval into an element of the preference set.*

We can now state formally the definition of a STPPU, which combines preferences from the definition of a STPP with contingency from the definition of a STPU. Note that we consider links that are hard constraints to be soft constraints with maximal preference.

Definition 2 (STPPU) *A* Simple Temporal Problem with Preferences and Uncertainty (STPPU) *is a tuple $P = (N_e, N_c, L_r, L_c, S)$ where:*

- *N_e is the set of executable timepoints;*
- *N_c is the set of contingent timepoints;*
- *L_r is the set of soft requirement constraints over S;*
- *L_c is the set of soft contingent constraints over S;*

- $S = \langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ *is a c-semiring.*

In order to analyse the solutions to a STPPU, we need some further preliminary definitions:

Definition 3
- requirement duration γ_{ij} *is any point in interval* I_{ij} *of requirement constraint* r_{ij}*, i.e.* $\gamma_{ij} = \gamma(t_j) - \gamma(t_i)$*;*
- contingent duration ω_{ij} *is any point in interval* $\hat{I}_{ij}$ *of contingent constraint* g_{ij}*, i.e.* $\omega_{ij} = \omega(e_j) - \delta(b_i)$*;*
- control sequence δ *of the STPPU an assignment of the executable time points* $\delta = \{\delta(b_1), \ldots, \delta(b_n)\}$*; if it is an assignment to all the executable timepoints, the control sequence is said to be* complete*, otherwise* partial*; every control sequence is associated with a preference value,* $\mathrm{pref}(\delta) = \prod_{r_{ij}|\exists\delta(b_i),\delta(b_j)} f_{ij}(\delta(b_j) - \delta(b_i))$*, where* $\prod$ *represents the multiplicative operator of the semiring;*
- space of complete situations *of the STPPU is the Cartesian product of all contingent interval* $\Omega = [\hat{l}_1, \hat{u}_1] \times \ldots \times [\hat{l}_G, \hat{u}_G]$*;*
- situation *(or* realisation*)* $\omega = \omega_1, \ldots, \omega_G$ *is an element of* Ω*; just like for a control sequence it can be complete or partial; every situation is associated with a preference* $\mathrm{pref}(w) = \prod_{i,j|\omega_{ij}\in\omega} \hat{f}_{ij}(\omega_{ij})$*;*
- *for all* $\omega \in \Omega$*, the* projection P_ω *of STPPU* P *is the STPP obtained replacing in all soft contingent constraints* g_k*,* $\hat{I}_k$ *with* $[w_k, w_k]$*.*

We say a *schedule* T is a complete assignment to the timepoints; every schedule has a preference value, $\mathrm{pref}(T)$:

Definition 4 (Schedule) *A* schedule T *is a complete assignment to all the timepoints of STPPU* P*; a schedule identifies an assignment* γ_T *or, more precisely, a control sequence* δ_T *and a situation* $w_T = \{w^T_{ij} | w^T_{ij} = \omega_T(e_j) - \delta_T(b_i)\}$ *(we will write* $T = (\delta_T, \omega_T)$*). Hence a schedule identifies a unique set of requirement durations* $\{\gamma^T_{ij} | \gamma^T_{ij} = \gamma_T(t_j) - \gamma_T(t_i)\}$ *and it is said to be* consistent *if* $\forall r_{ij}, f_{ij}(\gamma^T_{ij}) > 0$ *and* $\forall g_{ij}, \hat{f}_{ij}(\omega^T_{ij}) > 0$*. Every schedule is associated with a preference, simply* $\mathrm{pref}(T) = (\prod_{ij|\exists r_{ij}} f_{ij}(\gamma^T_{ij})) \times \mathrm{pref}(w_T)$*.*

We can now give two different types of controllability which take into account both contingency and preferences.

Definition 5 (Optimal Strong Controllability) *A STPPU is* Optimally Strongly Controllable *iff there exists a control sequence* δ *such that for all* $\omega \in \Omega$*,* $T = (\delta, \omega)$ *is a consistent schedule for* P_ω*, and* $\mathrm{pref}(T)$ *is optimal (i.e. there is no other schedule* T' *consistent with projection* P_ω *such that* $\mathrm{pref}(T') > \mathrm{pref}(T)$*).*

Definition 6 (α-Strong Controllability) *A STPPU is* α-Strongly Controllable*, with* $\alpha \in A$*, iff there exists a control sequence* δ *such that for all* $\omega \in \Omega$*,* $T = (\delta, \omega)$ *is a consistent schedule for projection* P_ω*, and* $\mathrm{pref}(T) \geq \alpha$*.*

Definition 7 (Optimal Weak Controllability) *A STPPU is* Optimally Weakly Controllable *iff for all* $\omega \in \Omega$ *there exists a control sequence* δ_ω *such that* $T = (\delta_\omega, \omega)$ *is a consistent schedule for projection* P_ω*, and* $\mathrm{pref}(T)$ *is optimal for* P_ω*.*

Definition 8 (α-Weak Controllability) *A STPPU is* α-Weakly Controllable*, with* $\alpha \in A$*, iff for all* $\omega \in \Omega$ *there exists a control sequence* δ_ω *such that* $T = (\delta_\omega, \omega)$ *is a consistent schedule for projection* P_ω*, and* $\mathrm{pref}(T) \geq \alpha$*.*

Checking Optimal Strong Controllability and α-Strong Controllability

In this section we describe an algorithm that checks, in polynomial time, whether a STPPU P is Optimally Strongly Controllable. The algorithm we propose relies on two known algorithms. The first is Path-Solver (Rossi *et al.* 2002a) which enforces path consistency on a STPP. The second is Strong-Controllability (Vidal & Ghallab 1996), which checks if a STPU is Strongly Controllable. The main idea is to apply Strong-Controllability to a special STPU, which we will call P_{opt}, that is constructed from the STPP P' obtained by first applying Path-Solver to STPPU P.

In order to able to use the former algorithms we need to impose a restriction on the shape of the preference functions, namely semi-convexity (Rossi *et al.* 2002a). (Recall that a function $f : I \to A$ is semi-convex if $\forall\alpha \in A$ the set of elements $\{x \in I | f(x) \geq \alpha\}$ forms a unique interval.) We will also assume that the semiring underlying our constraint problems is the *fuzzy semiring* $S_{\mathrm{FCSP}} = \langle [0,1], max, min, 0, 1\rangle$.

Any STPPU can be treated as a STPP ignoring the fact that some constraints are contingent. In particular we can consider function IU ('Ignore Uncertainty') that maps a STPPU $P = (N_e, N_c, L_r, L_c, S)$ into STPP $IU(P) = \langle I, f\rangle$ (Rossi *et al.* 2002a) where the set of intervals I is the set of all the intervals of soft constraints in L_r and L_c, and preference function $f : I \to A$ acts on each interval as the preference function of the soft constraint in P.

Now for checking Strong Controllability, since we are not interested in retrieving an actual solution, we only need to apply the first part of Path-Solver. We will call this sub-algorithm Soft-PC-2. Soft-PC-2 takes as input a STPP and enforces path consistency. As a result, it squeezes some intervals and lowers some preference functions. At the end, all the preference functions reach the same maximum preference level, which we will call opt, which corresponds to some sub-intervals of I.

Soft-PC-2 returns a STPP that has interesting features. First, the intervals consist of a minimal STP (i.e. a problem containing only points that appear in at least one solution). Second, the sub-STP consisting of the sub-intervals mapped by the preference functions into *opt* is minimal as well, and all its solutions are optimal solutions of the original STPP.

We will call P_{opt} the STPU obtained considering the sub-intervals mapped into opt on all the requirement constraints after Soft-PC-2, and the original intervals on all the contingent constraints. Notice that the semi-convexity of the preference functions guarantees that P_{opt} is a STPU and not a TCSPU. The procedure that, given as input a path consistent STPP, returns a STPU, with the structure we have just described will be referred to as OPT.

Of course, if any contingent constraint is squeezed during the enforcement of path consistency, we can conclude that the problem is not pseudo-controllable (Morris & Muscettola 2000) and hence not Strongly Controllable. Furthermore, the following theorem allows us to conclude that it cannot be Optimally Strongly Controllable.

Theorem 1 *If a STPPU P is Optimally Strongly Controllable (OSC) then the STPU Q, obtained simply ignoring preference functions on all the constraints, is Strongly Controllable (SC). However the converse does not hold.*

All proofs have been omitted for lack of space.

It can be shown that a STPPU P, with semi-convex functions, is OSC iff P_{opt} is SC, as the following theorem states:

Theorem 2 *STPPU P, with semi-convex preference functions, is Optimally Strongly Controllable iff the corresponding STPU P_{opt} is Strongly Controllable.*

To summarise, the algorithm we propose for checking Optimal Strong consistency of a STPPU P first applies Soft-PC-2 to $IU(P)$. If any contingent interval is squeezed during the process then the algorithm stops since the problem cannot be OSC. Otherwise it extracts P_{opt} from path consistent $IU(P)$, and runs Strong-Controllability on P_{opt}. The algorithm is shown in Figure 1.

Pseudocode for Path-OSC
1. **input** STPPU P;
2. STPP $J \leftarrow IU(P)$;
3. STPP $K \leftarrow$ Soft-PC-2 (J);
4. **if** any contingent is squeezed: return FALSE;
5. **else**:
6. STPU $P_{opt} \leftarrow OPT(K)$;
7. **if** Strong-Controllability (P_{opt}): return TRUE;
8. **else** return FALSE;

Figure 1: Checking OSC using Soft-PC-2.

Another possibility is to combine Strong-Controllability with Chop-Solver (Rossi *et al.* 2002a). Again, we are not interested in a actual solution so we will consider an algorithm very similar to Chop-Solver, Chop-PC-2. Recall that Chop-Solver performs a binary search of preference levels. At each level the STPP is 'chopped', meaning that only subintervals mapped into preferences equal or higher than the chopping level are kept and form a STP. At each level the consistency of the STP obtained by chopping is considered. Soft-PC-2 returns the consistent STP, STP_{opt}, corresponding to the highest level at which chopping leads to a consistent problem. At this point a procedure $\widehat{OPT}$, very similar to OPT, takes as input STP_{opt} and replaces all the intervals that originally belonged to contingent constraints with their original intervals, returning a STPU P_{opt}. Finally, Strong-Controllability is given P_{opt} as input. If P_{opt} is SC then P is OSC, otherwise P is not OSC. Figure 2 shows the pseudocode for this algorithm.

Both algorithms we propose are polynomial. The complexity of Soft-PC-2 and Chop-PC-2 is the same as Path-Solver and Chop-Solver (since finding an actual solution was not relevant in terms of complexity): $\mathcal{O}(n^3 \times R \times l)$, where $n = |N_r| + |N_c|$, R is the maximum range of an interval, and l is the number of preference levels. Procedures IU, OPT and $\widehat{OPT}$ are linear in the total number of constraints, which in turn is $\mathcal{O}(n^2)$. The complexity of Strong-Controllability is the same as the complexity of PC-2, i.e. $\mathcal{O}(n^3 \times R)$. We can conclude that both Path-OSC and Chop-OSC have a total complexity of $\mathcal{O}(n^3 \times R \times l)$. Note that this is in line with results on STPUs (Vidal & Ghallab 1996). In fact, just like SC for STPUs, the complexity of checking OSC of a STPPU has the same complexity of enforcing path consistency.

Pseudocode for Chop-OSC
1. **input** STPPU P;
2. STPP $J \leftarrow IU(P)$;
3. STP $STP_{opt} \leftarrow$ Chop-PC-2 (J);
4. STPU $P_{opt} \leftarrow \widehat{OPT}(STP_{opt})$;
5. **if** Strong-Controllability (P_{opt}): return TRUE;
6. **else** return FALSE;

Figure 2: Checking OSC using Chop-Solver.

α-Strong Controllability. We now tackle the problem of verifying whether a STPPU P is α-SC or not. First of all, let us point out the main difference between α-SC and OSC.

It is tempting to think that OSC is equivalent to *opt*-SC (i.e. α-SC with $\alpha = opt$, where *opt* is the maximum preference level at which Chop-Solver finds a consistent STP). However this is not the case. Both properties, OSC and α-SC, entail restrictions on the global preference associated with a schedule. OSC entails the existence of a control sequence that, when completed with a situation, is optimal for the projection corresponding to that situation. α-SC, however, imposes that the completed control sequence must have a preference at least α on all the projections.

For example, no STPPU can ever be α-consistent for any $\alpha > \alpha^* = min_{k \in \mathrm{Ctg}} \hat{f}_k(\omega_k)$. Indeed, suppose ω is a situation for which some constraint has preference smaller than α. Then a projection corresponding to ω has only solutions with preference strictly less than α.

Having said this, we will always consider $\alpha \leq \alpha^*$. The following theorem, similar to the one given for OSC, relates α-SC to SC.

Theorem 3 *If a STPPU P is α-Strongly Controllable (α-SC) then the STPU Q, obtained simply ignoring preference functions on all the constraints, is Strongly Controllable (SC). However the converse does not hold.*

It is possible to put α-SC of a STPPU P in one-to-one correspondence to SC of a related STPU P^α. P^α is the problem obtained chopping P at level α, as described in the previous section. Note that since $\alpha \leq \alpha^*$, contingent constraints maintain their intervals after the chop.

Theorem 4 *A STPPU P is α-Strongly Controllable, with $\alpha \leq \alpha^*$, iff the corresponding STPU P^α is Strongly Controllable.*

Figure 3 shows an algorithm to check α-SC. Procedure Chop(STPP, *pref*) takes as input a STPP and a preference level, e.g. α, and returns the STP obtained considering only intervals mapped by the preference functions to at least *pref*. Procedure Add-U(STP) adds the information of which constraints and points of the STP are to be considered contingent, hence transforming it in STPU.

Pseudocode for α-SC
1. **input** STPPU P;
2. STPP $J \leftarrow IU(P)$;
3. STP $J^\alpha \leftarrow$ Chop(J, α);
4. $P^\alpha \leftarrow$ Add-U(J^α);
5. **if** Strong-Controllability (P_α): return TRUE;
6. **else** return FALSE;

Figure 3: Checking α-Strong Controllability.

Another query one might want to answer is: what is the highest level α at which P is α-SC? We propose an algorithm that is very similar to Chop-Solver in the sense that the only modification is to replace, at every chop level, PC-2 with Strong-Controllability. Specifically, it is possible to define an algorithm Max-α-SC that performs a binary search for the highest level α at which the problem is α-SC. After chopping at level l the STP J^l obtained is transformed by function Add-U(STP) into STPU P^l and then passed to Strong-Controllability. Note that in general $\alpha \ll opt$.

The complexity of algorithm α-SC is clearly the same as the complexity of Strong-Controllability. In fact, the procedures of lines 2–4 are linear in the total number of constraints. The complexity of Max-α-SC is also tied to the complexity of Strong-Controllability. The algorithm itself consist of applying Strong-Controllability a number of times, at most polynomial in the number of nodes, as specified by the parameter *precision* given as input. We can conclude that the complexity of α-SC is $\mathcal{O}(n^3 \times R)$, and that of Max-$\alpha$-SC is $\mathcal{O}(p \times n^3 \times R)$, where p is proportional to the search precision required by the user.

Checking Optimal Weak Controllability and α-Weak Controllability

We now consider the impact of adding preferences with respect to the issue of Weak Controllability. The following theorem states how OWC and WC are related.

Theorem 5 *A STPPU P is Optimally Weakly Controllable (OWC) if the STPU Q, obtained simply ignoring preference functions on all the constraints, is Weakly Controllable (WC). However the converse does not hold.*

The converse fails in general because the theorem takes in account the possibility of mapping some elements of the intervals into 0. However if all the elements are mapped into strictly positive preferences, then the correspondence becomes biuniform:

Theorem 6 *A STPPU P, with strictly positive preference functions, is Optimally Weakly Controllable (OWC) iff the STPU Q, obtained simply ignoring preference functions on all the constraints, is Weakly Controllable (WC).*

This allows us to conclude that to check OWC, it is enough to apply algorithm Weak-Controllability proposed in Vidal & Ghallab (1996). Notice that the condition on preference functions in Theorem 6 is sufficient but not necessary.

Now to check α-WC we have two different approaches. The first approach is to chop the STPPU at level α and then to apply Weak-Controllability to the STPU obtained. The second possibility is to use the fact that a STPU is WC iff all the projections P_ω with $\omega \in \{\hat{l}_1, \hat{u}_1\} \times \ldots \times \{\hat{l}_h, \hat{u}_h\}$, where h is the number of contingent constraints, are consistent STPs (Vidal & Ghallab 1996). Using this, the second approach is to chop each projection P_ω at level α and then to check the consistency of the derived STP. The complexity of both algorithms is exponential in the number of contingent constraints h: $\mathcal{O}(2^h \times n^3 \times R)$.

Future Work

We have introduced the Simple Temporal Problem with Preferences and Uncertainty, and discussed Strong and Weak Controllability, together with algorithms to verify these properties. We would like to extend the concept of Dynamic Controllability to STPPUs in the same way, and develop methods to verify and execute a DC STPPU.

Acknowledgements. We would like to thank our supervisors, Francesca Rossi (University of Padova) and Carmen Gervet (Imperial College London), for collaborating with us on this research.

References

Bistarelli, S.; Montanari, U.; and Rossi, F. 1997. Semiring-based constraint solving and optimization. *Journal of the ACM* 44(2):201–236.

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

Khatib, L.; Morris, P.; Morris, R. A.; and Rossi, F. 2001. Temporal constraint reasoning with preferences. In *Proc. of IJCAI'01*, 322–327.

Morris, P., and Muscettola, N. 2000. Execution of temporal plans with uncertainty. In *Proc. of AAAI-2000*, 491–496.

Morris, P.; Muscettola, N.; and Vidal, T. 2001. Dynamic control of plans with temporal uncertainty. In *Proc. of IJCAI'01*, 494–502.

Rossi, F.; Sperduti, A.; Venable, K.; Khatib, L.; Morris, P.; and Morris, R. 2002a. Learning and solving soft temporal constraints: An experimental study. In *Proc. of CP'02*, 249–263.

Rossi, F.; Venable, K.; Khatib, L.; Morris, P.; and Morris, R. 2002b. Two solvers for tractable temporal constraints with preferences. In *Proc. of AAAI-02 Workshop on Preference in AI and CP*.

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

Vidal, T., and Ghallab, M. 1996. Dealing with uncertain durations in temporal constraint networks dedicated to planning. In *Proc. of ECAI-96*, 48–52.