Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Simple Temporal Problems with Taboo Regions

T. K. Satish Kumar[*]
Computer Science Department
University of Southern California
California, USA
tkskwork@gmail.com

Marcello Cirillo
AASS Research Centre
Örebro University
Sweden
marcello.cirillo@aass.oru.se

Sven Koenig
Computer Science Department
University of Southern California
California, USA
skoenig@usc.edu

Abstract

In this paper, we define and study the general framework of Simple Temporal Problems with Taboo regions (STPTs) and show how these problems capture metric temporal reasoning aspects which are common to many real-world applications. STPTs encode simple temporal constraints between events and user-defined taboo regions on the timeline, during which no event is allowed to take place. We discuss two different variants of STPTs. The first one deals with (instantaneous) events, while the second one allows for (durative) processes. We also provide polynomial-time algorithms for solving them. If all events or processes cannot be scheduled outside of the taboo regions, one needs to define and reason about "soft" STPTs. We show that even "soft" STPTs can be solved in polynomial time, using reductions to max-flow problems. The resulting algorithms allow for incremental computations, which is important for the successful application of our approach in real-time domains.

Introduction

Efficient algorithms for temporal reasoning are critical for a large number of real-world applications. Autonomous space exploration (Knight et al. 2001), domestic activity management (Pecora and Cirillo 2009) and job scheduling on servers (Ji, He, and Cheng 2007) are just a few applications which, although apparently different, require similar techniques for sophisticated and efficient temporal reasoning.

Many formalisms have been proposed and are currently used for reasoning with metric time, with varying degrees of complexity and expressiveness. *Simple Temporal Problems* (STPs) are on the lower end of the scale with respect to complexity. An STP can be encoded as a graph $G = \langle \mathcal{X}, \mathcal{E} \rangle$, where $\mathcal{X} = \{X_0, X_1 \cdots X_N\}$ is the set of vertices and $\mathcal{E}$ is the set of edges. Each $X_i \in \mathcal{X}$ represents an event, where X_0 conventionally represents the "beginning of time" and is set to 0. Each $e = \langle X_i, X_j \rangle \in \mathcal{E}$, annotated with the bounds $[LB(e), UB(e)]$, is a *simple temporal constraint* between X_i and X_j, indicating that X_j must be scheduled between $LB(e)$ and $UB(e)$ time units after X_i ($LB(e) \leq UB(e)$). Although their expressiveness is limited compared to other formalisms, STPs are widely used, as they can be solved in polynomial time using shortest path computations on their *distance graph* representations. In the distance graph representation, the constraint $X_j - X_i \leq w$ is represented as an edge from X_i to X_j annotated with w. The absence of negative cost cycles in the distance graph characterizes the consistency of the temporal constraints (Dechter, Meiri, and Pearl 1991). Shortest paths in the distance graph are commonly calculated using the Bellman-Ford algorithm. However, more recent and more efficient algorithms can be employed for solving STP instances with additional structure (Planken, De Weerdt, and van der Krogt 2008).

Disjunctive Temporal Problems (DTPs) are significantly more expressive than STPs, as they can encode disjunctive constraints. They can be used to model a large variety of real-world problems, such as scheduling problems with positive and negative time lags (Brucker, Hilbig, and Hurink 1999). A DTP is defined by a set of events $\mathcal{X} = \{X_0, X_1 \cdots X_N\}$ (as in the case of STPs, X_0 represents the "beginning of time") and a set of disjunctive constraints $\mathcal{C}$, where a constraint $c_i \in \mathcal{C}$ is a disjunction of the form $s_{(i,1)} \vee s_{(i,2)} \cdots s_{(i,Q_i)}$. Each disjunct $s_{(i,j)}$ ($1 \leq j \leq Q_i$) encodes a simple temporal constraint of the form $L_{(i,j)} \leq X_{b_{(i,j)}} - X_{a_{(i,j)}} \leq U_{(i,j)}$ ($0 \leq a_{(i,j)}, b_{(i,j)} \leq N$). Unfortunately, although DTPs are sufficiently expressive for most real-world applications, one needs an exponential search space to solve them.[1] The most common approach for solving DTPs is to convert the original problem to one of selecting a set of disjuncts, one from each constraint, which induce a consistent STP. While checking the consistency and finding a solution of an STP requires polynomial time, there is an exponential number of disjunct combinations to be tested. This "disjunct selection problem" can be cast as a Constraint Satisfaction Problem (CSP) (Oddi and Cesta 2000) or as a SATisfiability problem (SAT) (Armando, Castellini, and Giunchiglia 2000) and then be solved with the respective search procedures. DTPs are also efficiently solved using a circuit-based SAT encoding (Nelson and Kumar 2008).

[*]Alias: Satish Kumar Thittamaranahalli

[1]Many less expressive subsets of DTPs are also NP-hard to solve. A notable example is *Temporal Constraint Satisfaction Problems* (TCSPs). Here, disjunctions are limited to the form $X_j - X_i \in I_1 \cup I_2 \cdots I_k$, where the right-hand side encodes a union of disjoint intervals.

In this paper, we define Simple Temporal Problems with Taboo regions (STPTs), a formalism which allows the specification of a limited yet useful subset of DTPs, and we show how these problems can be solved in polynomial time. The formal definition of an STPT is based on that of an STP, that is, it is encoded by a graph $G = \langle \mathcal{X}, \mathcal{E} \rangle$, where $\mathcal{X}$ is a set of events and $\mathcal{E}$ is the set of edges in the form of simple temporal constraints. The definition of an STPT is completed by a set of *taboo regions*, which is a set of M temporal regions in which no event can be scheduled, defined by $\mathcal{T} = \{T_1, T_2 \cdots T_M\}$. Each $T_i = (a_i, b_i) \in \mathcal{T}$ is a time region specified by an *open interval* with left end point a_i and right end point b_i, henceforth called the starting and ending time points, respectively. STPTs can represent problems where one needs to schedule events while respecting forbidden time regions. Taboo regions are not necessarily related to the events themselves. They could, for example, be a consequence of external constraints arising from the agent or the environment and be used to encode maintenance periods in a server job-scheduling setting, sleeping periods for the CPU on a mobile device or periods of forced inactivity for mobile robots or rovers. Consider a planetary rover which relies on solar power to perform its operations. In this case, there can exist time constraints between tasks as well as time intervals during which no task can be executed, for example, due to low battery levels. These time intervals can be represented as taboo regions in our formalism and the overall problem can be cast as an STPT.

In the remainder of this paper, we describe two variants of STPTs in more detail: the first one with (instantaneous) events and the second one with (durative) processes. For each variant, we define the vanilla ("hard") version as well as the "soft" version. We show how the vanilla version of both variants can be easily reduced to a known tractable class of DTPs. We reduce the "soft" version of STPTs with events to a known tractable class of Simple Temporal Problems with Preferences (STPPs). Finally, we describe a novel reduction from the "soft" version of STPTs with processes to max-flow problems on bipartite graphs. The solution techniques for both "soft" versions rely on max-flow algorithms, which are known to be amenable to incremental computations.

STPTs with Events

The first variant of STPTs that we define is with (instantaneous) events. It has the same definition as provided in the Introduction and is characterized by $G = \langle \mathcal{X}, \mathcal{E} \rangle$ and $\mathcal{T} = \{T_1, T_2 \cdots T_M\}$, where each $T_i = (a_i, b_i) \in \mathcal{T}$ is a time region specified by an open interval with left end point a_i and right end point b_i. This variant is useful in situations where the execution times of events are at least an order of magnitude smaller than the granularity at which the temporal constraints are specified.

We assume, without loss of generality, that the taboo regions are specified in a *canonical form*, namely that $a_1 < b_1 \leq a_2 < b_2 \leq a_3 \cdots b_M$ holds. This means that there are no overlaps between different taboo regions. If overlaps exist, they can be eliminated by merging overlapping taboo regions. A canonical form can then be obtained by arranging the non-overlapping taboo regions in ascending order.

Compliance with taboo regions can be expressed by N constraints of the form $X_i \notin T_1 \cup T_2 \cdots T_M$, where $1 \leq i \leq N$. Because $T_1, T_2 \cdots T_M$ are assumed to be in canonical form, this constraint can be translated to $X_i - X_0 \in (-\infty, a_1] \cup [b_1, a_2] \cdots [b_M, \infty)$. Therefore, the overall STPT contains $|\mathcal{E}|$ simple temporal constraints and N disjunctive constraints of the above nature, each with $M + 1$ disjuncts.

There are two ways how one can prove the tractability of these variants of STPTs: the first one is to reduce them to *Restricted Disjunctive Temporal Problems* (RDTPs) (Kumar 2005a), while the second one is to reduce them to a known tractable subclass of *Simple Temporal Problems with Preferences* (STPPs) (Kumar 2004).

Reducibility to RDTPs

RDTPs were first introduced in (Kumar 2005a). They are a restricted but highly expressive class of DTPs that allows for disjunctions in temporal constraints while still maintaining tractability. They are characterized by a set of events $\mathcal{X} = \{X_0, X_1 \cdots X_N\}$ (where X_0, as in the case of STPs, represents the "beginning of time") and a set of constraints $\mathcal{C}$. A constraint $c_i \in \mathcal{C}$ is of one of three types:

(Type 1) $L \leq X_b - X_a \leq U$

(Type 2) $(L_1 \leq X_a \leq U_1) \vee (L_2 \leq X_a \leq U_2) \cdots (L_{Q_i} \leq X_a \leq U_{Q_i})$

(Type 3) $(L_1 \leq X_a \leq U_1) \vee (L_2 \leq X_b \leq U_2)$

Let $\mathcal{C} = \mathcal{E} \cup \mathcal{C}'$, where $\mathcal{E}$ is the set of simple temporal Type 1 constraints and $\mathcal{C}'$ is the set of disjunctive temporal Type 2 and Type 3 constraints. RDTPs can be solved in polynomial time by converting them to meta-level CSPs with Connected Row Convex (CRC) constraints (Kumar 2005a). The complexity of the first step in the conversion is $O((N'+1)N|\mathcal{E}|)$, where N' is the number of variables that occur in any of the Type 2 or Type 3 constraints. The resulting meta-level CSP has $\mathcal{C}'$ variables with maximum domain size equal to Q, where Q is defined to be the maximum number of disjuncts in any Type 2 or Type 3 constraint. Because the complexity of solving CRC constraints on V variables with maximum domain size D is $O(V^3 D^2)$ (Kumar 2005b), the total complexity of solving RDTPs is $O((N' + 1)N|\mathcal{E}| + |\mathcal{C}'|^3 Q^2)$.

STPTs with events can be readily cast into equivalent RDTPs because the STP core of a given STPT (that is, the $|\mathcal{E}|$ simple temporal constraints) fits the Type 1 constraints of RDTPs, while the N disjunctive constraints of the STPT fit the Type 2 constraints with $M + 1$ disjuncts each. Therefore, STPTs with events can be solved in time $O(N^2|\mathcal{E}| + N^3 M^2)$ by virtue of this reduction.

Reducibility to STPPs

STPPs have been studied in the context of "soft" constraint satisfaction (Khatib et al. 2001). Although STPPs are NP-hard to solve in general, interesting subclasses have been identified which are solvable in polynomial time. One such subclass is presented in (Kumar 2004). Here, we are given a simple temporal network $G = \langle \mathcal{X}, \mathcal{E} \rangle$ with a family of piecewise constant preference functions $F = \{f_{X_i}(t) : R \rightarrow R\}$. The function $f_{X_i}(t)$ specifies the preference of scheduling

$X_i \in \mathcal{X}$ at time t. The goal is to produce a schedule that satisfies all temporal constraints and maximizes the sum of the preferences associated with the events, called the *total preference value*. This subclass of STPPs (where each preference function is associated with a single event) is solvable in time $O(N^2|\mathcal{E}| + N^{2.5}I^{2.5})$, where I is the maximum number of intervals defined by any of the piecewise constant preference functions (Kumar 2004). The first term $N^2|\mathcal{E}|$ accounts for the complexity of reducing the STPP to a Partially Ordered SET (POSET), while the second term $N^{2.5}I^{2.5}$ accounts for the complexity of computing the *largest weighted anti-chain* on this POSET by using max-flow techniques on bipartite graphs (Cormen et al. 2001).

With an appropriate choice of piecewise constant preference functions, we can translate STPTs into this subclass of STPPs. The function $f_{X_i}(t)$ is the same for every X_i:

$$f_{X_i}(t) = \begin{cases} 1 & \text{if } t \in (-\infty, a_1] \cup [b_1, a_2] \cdots [b_M, \infty) \\ 0 & \text{otherwise} \end{cases}$$

Theorem 1. *A solution of a given STPT with events is characterized by a total preference value of N for its corresponding STPP.*

Proof. If event $X_i \in \mathcal{X}$ is scheduled outside the taboo regions in $\mathcal{T}$, then the corresponding preference function f_{X_i} yields a value of 1. Therefore the total preference value for a solution of the STPT is equal to N. Conversely, since no preference function yields a value greater than 1, the only possible way to obtain a total preference value of N is to schedule every event outside the taboo regions. $\square$

It is straightforward to see that, using this approach, STPTs can be solved in time $O(N^2|\mathcal{E}| + N^{2.5}M^{2.5})$, as the maximum number of intervals defined by any preference function is equal to $M + 1$. The reduction of an STPT with events to its equivalent STPP of this subclass allows us to exploit properties of existing algorithms for solving the latter. The first benefit is that the STPT formalism can be generalized to a "soft" optimization version when no solution of the "hard" version exists. The second benefit is that both versions of the problem allow for incremental computations.

Optimization Version of STPTs with Events

It might not always be possible to find a schedule which accommodates all events outside the taboo regions. We thus also define a "soft" version of STPTs which aims at finding a schedule that maximizes the number of events scheduled outside the taboo regions. This version of STPTs is also captured by the subclass of STPPs with the same definition of piecewise constant preference functions f_{X_i} given above.

Moreover, by using different scaling factors for different f_{X_i}, we can maximize the number of events scheduled outside the taboo regions weighted by their *priorities*.[2] For each event X_i with priority p_i, the associated f_{X_i} therefore is:

[2]We adopt the convention that the importance of an event is directly proportional to its priority, which we use as its scaling factor.

$$f_{X_i}(t) = \begin{cases} p_i & \text{if } t \in (-\infty, a_1] \cup [b_1, a_2] \cdots [b_M, \infty) \\ 0 & \text{otherwise} \end{cases}$$

Incremental Computations

Because the solution procedure for STPTs with events entailed by this reduction involves computing the max-flow on bipartite graphs, it allows for incremental computations. This means that, if we solve an instance of the STPT which is later updated, we can reuse the computation for solving the original instance. The complexity of this incremental algorithm depends only on the parameters which characterize the difference between the original instance and the updated one. The reduction of an STPT with events to a max-flow problem requires the computation of shortest paths, which can be made incremental, as shown in (Kumar 2003). The solutions of max-flow instances can also be computed incrementally, as shown in the same paper. Put together, STPTs with events can be solved incrementally, which is important for the applicability of this formalism to real-time domains.

STPTs with Processes

We now define a second variant of STPTs, where we do not consider events but rather *processes*, which, by definition, have durations. In our case, the durations are assumed to be controllable within specified bounds. Here, we show how to incorporate such processes into our framework, yielding the same representation we formalized above.

STPTs with processes are also characterized by $G = \langle \mathcal{X}, \mathcal{E} \rangle$ and $\mathcal{T} = \{T_1, T_2 \cdots T_M\}$. We represent processes by capturing their end points in $\mathcal{X}$ and their durations in $\mathcal{E}$ as follows. Given a set $\mathcal{P} = \{P_1, P_2 \cdots P_K\}$ of processes, each $P_i \in \mathcal{P}$ has a starting and an ending time point and a duration. The starting and ending time points can be represented as (instantaneous) events, while the duration can be formalized as a simple temporal constraint. Formally, we represent each P_i as a pair of events, namely $X^s_{P_i} \in \mathcal{X}$ representing its starting time point and $X^e_{P_i} \in \mathcal{X}$ representing its ending time point. An edge $\tilde{e}_{P_i} = \langle X^s_{P_i}, X^e_{P_i} \rangle \in \mathcal{E}$ is annotated with the bounds $[LB(\tilde{e}_{P_i}), UB(\tilde{e}_{P_i})]$ (where $UB(\tilde{e}_{P_i}) \geq LB(\tilde{e}_{P_i}) \geq 0$), that capture the flexibility in the duration of the process while ensuring that it is non-negative. $\mathcal{E}$ can also contain simple temporal constraints between the starting and ending time points of different processes.

A solution of an STPT instance with processes is a schedule in which no process intersects any taboo region. We assume that, once a process has started, it needs to be completed (that is, no process can be preempted). Since we assume the taboo regions to be in canonical form ($a_1 < b_1 \leq a_2 \cdots b_M$), each process P_i in a solution has to respect the disjunctive constraint:

$$(X^s_{P_i}, X^e_{P_i} \leq a_1) \vee (X^s_{P_i}, X^e_{P_i} \geq b_M) \vee \bigvee_{j=1}^{M-1} ((X^e_{P_i} \leq a_{j+1}) \wedge (X^s_{P_i} \geq b_j)),$$

that is, it should start and end either before the first taboo region, after the last one or between two consecutive taboo regions.

Reducibility to RDTPs

The reduction of STPTs with processes to their equivalent RDTPs is straightforward. The constraints that each process $P_i \in \mathcal{P}$ should be scheduled outside any taboo region $T_j \in \mathcal{T}$ can be formulated as the MK disjunctive constraints

$$(X^e_{P_i} \le a_j) \vee (X^s_{P_i} \ge b_j),$$

where $1 \le i \le K$ and $1 \le j \le M$. These MK constraints, one for each process-taboo region pair, represent the requirements specified above. One can immediately translate these constraints into the Type 3 constraints defined for RDTPs:

$$(-\infty < X^e_{P_i} \le a_j) \vee (b_j \le X^s_{P_i} < \infty).$$

We can now calculate the complexity of solving this version of STPTs with processes by examining the previously discussed complexity of solving RDTPs. In the current case, there are only $|\mathcal{E}|$ Type 1 constraints and MK Type 3 constraints. Since every Type 3 constraint has only 2 disjuncts, Q is equal to 2. Therefore, the "hard" version of STPTs with processes can be solved in time $O(NK|\mathcal{E}| + M^3K^3)$.

Optimization Version of STPTs with Processes

Just like for STPTs with events, we define a "soft" variant of STPTs with processes, as there are many practical situations in which some processes cannot be kept away from the taboo regions. We are therefore interested in scheduling as many processes as possible outside the taboo regions. Also, a process should intersect as few taboo regions as possible (see the example in Figure 1).

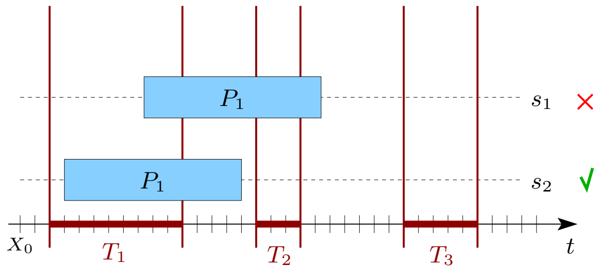

Figure 1: Illustration of the objective of our "soft" version of STPTs with processes. Whenever a process cannot be scheduled outside taboo regions, we are interested in minimizing the number of taboo regions it overlaps with. In the example above, assume that process P_1 must be scheduled between X_0 and the starting time point of taboo region T_3. Due to its duration, process P_1 will intersect *at least* one taboo region. We prefer schedule s_2 over schedule s_1, as process P_1 then intersects only taboo region T_1 instead of taboo regions T_1 and T_2.

One way to capture this objective is to minimize the number of overlapping (P_i, T_j) pairs. We can go further and associate penalties with (P_i, T_j) pairs. Then, our objective is:

$$\operatorname*{argmin}_s \sum_{i=1}^{K}\sum_{j=1}^{M} c_{ij} b_{ij}(s),$$

where $b_{ij}(s)$ is equal to 1 if process P_i and taboo region T_j intersect in schedule s and 0 otherwise, and $c_{ij} \ge 0$ is the penalty associated with the intersection. Of course, s is required to be a consistent schedule for the constraints in the STP core (that is, the set of all simple temporal constraints specified by $\mathcal{E}$). An equivalent objective is:

$$\operatorname*{argmax}_s \sum_{i=1}^{K}\sum_{j=1}^{M} c_{ij} \bar{b}_{ij}(s),$$

where $\bar{b}_{ij}$ is $1 - b_{ij}$.

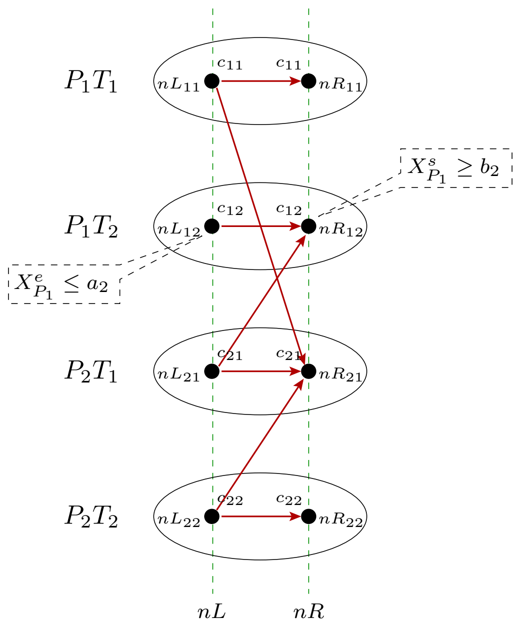

Figure 2: Example of a conflict graph, which encodes the size-2 conflicts that arise while attempting to schedule (P_i, T_j) pairs without overlaps. The size-1 conflicts are simply removed from the conflict graph, along with their accompanying edges. There are no higher-order conflicts. Since activating either an nL_{ij} or an nR_{ij} node for the same (P_i, T_j) pair contributes c_{ij} to the objective function and their activations are mutually exclusive, both nodes in each pair have weight c_{ij}.

This maximization problem can be encoded as an optimization problem on a graph. The graph encodes all possible conflicts in attempting to schedule (P_i, T_j) pairs without overlaps and is built as follows: Each pair (P_i, T_j) is encoded as two nodes (grouped in an oval in Figure 2). Each of the two nodes represents one of the two ways of ensuring that P_i and T_j do not overlap. One node is associated with the constraint $X^e_{P_i} \le a_j$, while the other one is associated with the constraint $X^s_{P_i} \ge b_j$ (since the process can either end before taboo region T_j or start after it). The overall graph is composed of $2KM$ nodes (grouped in KM ovals).

As shown in Figure 2, we call the nodes associated with (P_i, T_j) pairs nL_{ij} and nR_{ij}, where nL_{ij} encodes the constraint $X^e_{P_i} \leq a_j$ and nR_{ij} encodes the constraint $X^s_{P_i} \geq b_j$.

Starting from the STP core and compiling the constraints in it to the distance graph, we can now define the concept of the *activation* of a set of nodes.

Definition A set of nodes can be activated iff the constraints that they encode, when incorporated into the STP core, do not create an inconsistency.

The optimization problem is therefore reduced to activating as many nodes as possible, weighted by their penalties c_{ij}, without introducing any inconsistency. Nodes belonging to the same pair cannot be activated simultaneously.

Definition A *conflict* is a set of nodes all of which cannot be activated simultaneously. A *minimal conflict* is a conflict no proper subset of which is also a conflict.

The above definition implies that every conflict has a minimal conflict, and a set of nodes can therefore be activated simultaneously iff none of its subsets is a minimal conflict. The following Lemma proves the bounded nature of minimal conflicts. Central to its proof is the notion of *special edges*, which are the edges added to the STP core in an attempt to activate a set of nodes.

Lemma 1. *The size (or, synonymously, cardinality) of a minimal conflict is at most 2.*

Proof. Suppose we try to activate the set of nodes $\{nL_{i_1j_1}, nL_{i_2j_2} \cdots nL_{i_lj_l}\} \cup \{nR_{i_{l+1}j_{l+1}} \cdots nR_{i_rj_r}\}$. This involves the addition of the following outgoing and incoming edges with respect to X_0. The outgoing edges (of X_0) correspond to the nL nodes: the edge from X_0 to $X^e_{P_{i_1}}$ is annotated with a_{j_1}, and so on, up to the edge from X_0 to $X^e_{P_{i_l}}$. The incoming edges (of X_0) correspond to the nR nodes: the edge from $X^s_{P_{i_{l+1}}}$ to X_0 is annotated with $-b_{j_{l+1}}$, and so on, up to the edge from $X^s_{P_{i_r}}$ to X_0. We call these edges *special edges*. If the original distance graph contains one or more negative cost cycles, then the size of a minimal conflict is 0. Otherwise, a new negative cost cycle can occur only with the introduction of special edges and must therefore involve a special edge. Since all special edges have X_0 as an end point, any negative cost cycle must involve X_0. Thus, if there exists a negative cost cycle, then there also exists a negative cost cycle that contains X_0 only once and therefore contains at most two special edges. Since special edges correspond to the activation of nodes, the size of a minimal conflict is at most 2. □

Because the sizes of the minimal conflicts are bounded by a constant, we can enumerate all of them in polynomial time. A size-1 conflict can occur in two ways, namely

- because of a single outgoing special edge, if $dist(X^e_{P_i}, X_0) + a_j < 0$. In other words, imposing that process P_i should end before taboo region T_j is inconsistent with the STP core. An example is the special edge from X_0 to $X^e_{P_{i_1}}$ in Figure 3.

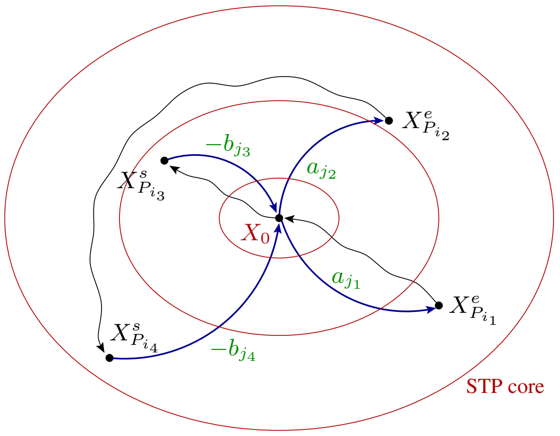

Figure 3: Illustration of the arguments used in proving the bounded sizes of minimal conflicts. Concentric circles visually represent the STP core. Curly edges represent shortest distances in the core, while the other edges are special edges that represent the induced constraints for activating nL and nR nodes and are labeled with appropriate bounds.

- because of a single incoming special edge, if $dist(X_0, X^s_{P_i}) - b_j < 0$. In other words, imposing that process P_i should start after taboo region T_j is inconsistent with the STP core. An example is the special edge from $X^s_{P_{i_3}}$ to X_0 in Figure 3.

A size-2 conflict can only occur with one incoming special edge and one outgoing special edge. In the context of Figure 3, consider a conflict involving the outgoing special edge from X_0 to $X^e_{P_{i_2}}$ and the incoming special edge from $X^s_{P_{i_4}}$ to X_0. The negative cost cycle in this conflict would be generated by $dist(X^e_{P_{i_2}}, X^s_{P_{i_4}}) + a_{j_2} - b_{j_4} < 0$, meaning that the constraints present in the STP core do not allow P_{i_2} to be scheduled before taboo region T_{j_2} and P_{i_4} to be scheduled after taboo region T_{j_4} simultaneously. This is so because the processes P_{i_2} and P_{i_4} are temporally related through constraints in the STP core and the added special edges would try to either "squeeze" them too close or "spread" them too far apart in time.

Because of Lemma 1, there are no higher-order conflicts. The size-2 conflicts can be modeled as edges between the corresponding nodes in the *conflict graph* (see Figure 2). The only other kind of conflicts, namely the size-1 conflicts, can be modeled by deleting the corresponding nodes in the conflict graph along with their accompanying edges.

Lemma 2. *The conflict graph is a bipartite graph, with the two partitions corresponding to the nL nodes and the nR nodes, respectively.*

Proof. Since size-2 conflicts always involve exactly one incoming edge, which corresponds to the activation of an nR node, and exactly one outgoing edge, which corresponds to the activation of an nL node, the edges in the conflict graph will always be between an nL node and an nR node but never between two nR nodes or two nL nodes. Therefore, the nL nodes and nR nodes can be separated into two partitions, resulting in a bipartite graph. □

We represent a size-2 conflict in the conflict graph by connecting two conflicting nodes by a directed edge. We adopt the convention of directing the edges from the node that contributes the outgoing edge in the distance graph to the one that contributes the incoming edge in the distance graph (see Figure 2). Also, as stated before, any nR_{ij} and nL_{ij} nodes are inconsistent with each other, as they would create a negative cost cycle if added simultaneously to the distance graph. This is reflected by a directed edge in the conflict graph from nL_{ij} to nR_{ij} for all $P_i \in \mathcal{P}$ and $T_j \in \mathcal{T}$.

Theorem 2. *The optimal solution of the "soft" version of STPTs with processes can be found in polynomial time.*

Proof. We build the conflict graph, which encodes all size-1 and size-2 conflicts. While nodes constituting size-1 conflicts are simply removed from the conflict graph with all their accompanying edges, size-2 conflicts are avoided by choosing an independent set. An independent set is a collection of nodes, no two of which are connected through an edge. A maximum weighted independent set therefore encodes an optimal activation of nodes. Since the conflict graph is bipartite, a maximum weighted independent set can be computed in polynomial time. We then compute the corresponding solution of the STPT with processes using standard shortest path computations in the distance graph of the STP core with added special edges corresponding to the maximum weighted independent set. □

Algorithm 1 presents the complete procedure for solving "soft" STPTs with processes by reducing them to max-flow instances on bipartite graphs. Because this algorithm is based on shortest path computations and max-flow, it can be made incremental as previously discussed.

We now analyze the complexity of solving STPTs with processes under this reduction. First, we have to compute the conflict graph. This can be achieved by computing the single-source shortest path trees with the Bellman-Ford algorithm $K + 1$ times, rooted at the $X^e_{P_i}$ nodes and the X_0 node, resulting in a complexity of $O(KN|\mathcal{E}|)$. Next, we have to determine the edges in the conflict graph, resulting in a complexity of $O(K^2M^2)$, given that we have to check for conflicts between any two pairs of a process and a taboo region. Then, we have to compute the max-flow on the bipartite conflict graph, resulting in a complexity of $O(K^{2.5}M^{2.5})$, as shown in (Cormen et al. 2001). Finally, we have to retrieve the solution by computing shortest paths on the new distance graph, where the number of edges could now have increased from $|\mathcal{E}|$ to $|\mathcal{E}| + MK$, resulting in a complexity of $O(N(|\mathcal{E}| + MK))$. The overall complexity of solving the "soft" version of STPTs with processes under this reduction is therefore $O(NK(M + |\mathcal{E}|) + K^{2.5}M^{2.5})$.

Algorithm 1: SOFT-STPTs-WITH-PROCESSES

```
Input: Simple Temporal Network $G = \langle \mathcal{X}, \mathcal{E} \rangle$;
       Taboo regions $\mathcal{T} = \{T_1, T_2 \cdots T_M\}$;
       Processes $\mathcal{P} = \{P_1, P_2 \cdots P_K\}$;
       Penalties $\{c_{ij} | 1 \le i \le K, 1 \le j \le M\}$
Output: Schedule $s^*$ with $s^* = \operatorname{argmin}_s \sum_{i=1}^{K} \sum_{j=1}^{M} c_{ij} b_{ij}(s)$,
        where $b_{ij}(s) = 1$ if process $P_i$ and taboo region $T_j$ intersect in
        schedule $s$ and 0 otherwise.
1 Construct the distance graph $\mathcal{D}(G)$ for $G = \langle \mathcal{X}, \mathcal{E} \rangle$
2 Construct the bipartite conflict graph encoding size-2 conflicts as follows:
      Nodes in the left-side partition are $nL_{ij}$ $(1 \le i \le K, 1 \le j \le M)$;
        the weight of $nL_{ij}$ is $c_{ij}$
      Nodes in the right-side partition are $nR_{ij}$ $(1 \le i \le K, 1 \le j \le M)$;
        the weight of $nR_{ij}$ is $c_{ij}$
      Add a directed edge from $nL_{i_1 j_1}$ to $nR_{i_2 j_2}$ iff
        $dist(X^e_{P_{i_1}}, X^s_{P_{i_2}}) + a_{j_1} - b_{j_2} < 0$ in $\mathcal{D}(G)$
3 Remove all size-1 conflicts as follows:
      Remove $nL_{ij}$ with its edges iff $dist(X^e_{P_i}, X_0) + a_j < 0$ in $\mathcal{D}(G)$
      Remove $nR_{ij}$ with its edges iff $dist(X_0, X^s_{P_i}) - b_j < 0$ in $\mathcal{D}(G)$
4 Compute maximum weighted independent set $\mathcal{W}$ on resulting conflict graph
5 Modify distance graph $\mathcal{D}(G)$ as follows:
      Add $e = \langle X_0, X^e_{P_i} \rangle$ with weight $a_j$ iff $nL_{ij} \in \mathcal{W}$
      Add $e = \langle X^s_{P_i}, X_0 \rangle$ with weight $-b_j$ iff $nR_{ij} \in \mathcal{W}$
6 Compute schedule $s^*$ as follows:
      Execution time of $X \in \mathcal{X} \leftarrow$ distance from $X_0$ to $X$ in the new distance
      graph
7 Return $s^*$
```

Conclusions

In this paper, we have introduced STPTs, a new class of temporal reasoning problems where we define taboo regions on top of an STP. Taboo regions capture externally imposed constraints on the schedulability of events or processes. We have defined two variants of STPTs: one with (instantaneous) events and one with (durative) processes.

We reduced both variants to known tractable classes of metric temporal reasoning problems. We went on to define "soft" versions of the considered STPTs. The "soft" version of STPTs with events turned out to be reducible to a known tractable class of STPPs, which made it possible to solve the original problem incrementally. The "soft" version of STPTs with processes required a novel reduction to max-flow problems on bipartite graphs. The nature of such flow problems once again made it possible for us to solve the original problem incrementally.

The class of problems studied in this paper is potentially very useful for solving many real-world tasks. In our future work, we intend to present concrete applications of our algorithms in real-world domains.

Acknowledgments.

This paper is based upon research supported by a MURI under contract/grant number N00014-09-1-1031. The research was performed while Marcello Cirillo visited the University of Southern California. His research visit was supported by the "Safe Autonomous Navigation" project (SAUNA), funded by the Swedish Knowledge Foundation (KKS). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies or the U.S. government.

References

Armando, A.; Castellini, C.; and Giunchiglia, E. 2000. SAT-based procedures for temporal reasoning. In *Proceedings of the European Conference on Planning (ECP)*.

Brucker, P.; Hilbig, T.; and Hurink, J. 1999. A branch and bound algorithm for a single-machine scheduling problem with positive and negative time-lags. *Discrete Applied Mathematics* 94(1):77–99.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2001. *Introduction to algorithms*. MIT press.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1):61–95.

Ji, M.; He, Y.; and Cheng, T. 2007. Single-machine scheduling with periodic maintenance to minimize makespan. *Computers & Operations Research* 34(6):1764–1770.

Khatib, L.; Morris, P.; Morris, R.; and Rossi, F. 2001. Temporal constraint reasoning with preferences. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Knight, R.; Rabideau, G.; Chien, S.; Engelhardt, B.; and Sherwood, R. 2001. Casper: Space exploration through continuous planning. *IEEE Intelligent Systems* 16(5):70–75.

Kumar, T. K. S. 2003. Incremental computation of resource-envelopes in producer-consumer models. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP)*.

Kumar, T. K. S. 2004. A polynomial-time algorithm for simple temporal problems with piecewise constant domain preference functions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Kumar, T. K. S. 2005a. On the tractability of restricted disjunctive temporal problems. In *Proceedings of the International Conference on Planning and Scheduling (ICAPS)*.

Kumar, T. K. S. 2005b. On the tractability of smooth constraint satisfaction problems. In *Proceedings of the Second International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR)*.

Nelson, B., and Kumar, T. K. S. 2008. CircuitTSAT: A solver for large instances of the disjunctive temporal problem. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*.

Oddi, A., and Cesta, A. 2000. Incremental forward checking for the disjunctive temporal problem. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*.

Pecora, F., and Cirillo, M. 2009. A constraint-based approach for plan management in intelligent environments. In *Proceedings of the Workshop on Scheduling and Planning Applications at ICAPS 2009*.

Planken, L.; De Weerdt, M.; and van der Krogt, R. 2008. P^3C: A new algorithm for the simple temporal problem. In *Proceedings of the International Conference on Planning and Scheduling (ICAPS)*.