

# Time-dependent Simple Temporal Networks: Properties and Algorithms

Cédric Pralet and Gérard Verfaillie

ONERA – The French Aerospace Lab

F-31055, Toulouse, France

cedric.pralet,gerard.verfaillie@onera.fr

## Abstract

Simple Temporal Networks (STNs) allow minimum and maximum distance constraints between time-points to be represented. They are often used when tackling planning and scheduling problems that involve temporal aspects. This paper is a summary of the journal article *Time-dependent Simple Temporal Networks: Properties and Algorithms* published in RAIRO - Operations Research. This journal article introduces an extension of STN called Time-dependent STN (TSTN), which covers temporal constraints for which the temporal distance required between two time-points is not necessarily constant. Such constraints are useful to model time-dependent scheduling problems, in which the duration of an activity may depend on its starting time. The paper introduces the TSTN framework, its properties, resolution techniques, as well as examples of applications.

## Motivation

Managing temporal aspects is crucial when tackling planning and scheduling problems. Indeed, such problems often involve constraints on the earliest start times and latest end times of activities, precedence constraints between activities, or no-overlapping constraints over sets of activities which use the same resource. More generally, they often involve constraints over the minimum and maximum temporal distance between time-points, the latter corresponding to the start or the end of activities.

A commonly used framework for handling such constraints is the framework of Simple Temporal Networks (STNs (Dechter, Meiri, and Pearl 1991)). This framework considers conjunctions of *simple temporal constraints*, which are constraints of the form  $x - y \in [a, b]$  with  $x, y$  two time-points and  $a, b$  two constants. STNs are appealing in practice due to the polynomial complexity of important operations such as computing the earliest/latest possible times associated with each time-point. STNs are also often used as a basic element when solving more complex temporal problems such as Disjunctive Temporal Networks (Stergiou and Koubarakis 2000).

In this work, we introduce an extension of STN called *Time-dependent STN* (TSTN (Pralet and Verfaillie 2013b)),

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

which covers temporal constraints for which the minimum and maximum distances required between two time-points  $x$  and  $y$  are not necessarily constant. More precisely, constraints manipulated in the TSTN framework take the form  $y - x \in [f(x, y), g(x, y)]$ . Such constraints are useful to model *time-dependent scheduling* problems (Cheng, Ding, and Lin 2004; Gawiejnowicz 2008), in which the duration of an activity may depend on its starting time, or problems in which the transition time required between two activities may depend on the time at which the transition is triggered. Such aspects are present in several domains:

1. in logistics, where the estimated time to go from location  $A$  to location  $B$  depends on the traffic congestion and thus on the starting time: constraint  $y - x \geq f(x)$  where  $x$  and  $y$  are the time-points associated with the start and the end of the move, and where  $f$  is a function such that  $f(t)$  gives the duration required to move from  $A$  to  $B$  by starting the move at time  $t$ ;
2. in production planning, where the duration required to perform a task at a given time depends on the resources available at that time (manpower, machines...); for instance, let  $x$  and  $y$  be the start and end times of a task, let  $E$  be the total amount of energy required for achieving this task, and let  $p(t)$  denote the manpower available at time  $t$ ; then, a task starting at time  $x$  needs duration  $f(t) = \min\{\delta \mid \int_t^{t+\delta} p(u) du \geq E\}$  for being achieved, which means that constraint  $y - x \geq f(x)$  must hold;
3. in the space domain, where *e.g.* the duration required by a satellite for moving from a pointing to Earth area  $A$  to a pointing to Earth area  $B$  depends on the time at which the transition from  $A$  to  $B$  occurs: constraint  $y - x \geq f(x, y)$  with  $x$  and  $y$  the start and the end of the move activity respectively, and with  $f$  a function such that  $f(t, t')$  gives the duration required to move from the pointing direction the satellite must have to point to  $A$  at time  $t$  to the pointing direction the satellite must have to point to  $B$  at time  $t'$ , taking into account the movement of the satellite on its orbit and the rotation of Earth on itself.

Constraints  $y - x \in [f(x, y), g(x, y)]$  manipulated in TSTN can always be rewritten as sets of inequality constraints  $y - x \geq dmin(x, y)$ . Informally,  $dmin(x, y)$  specifies a minimum temporal distance between the events associated with time-points  $x$  and  $y$ . Constraints  $y - x \geq$

$dmin(x, y)$  are called time-dependent simple temporal constraints, or more simply *t-simple temporal constraints*.

Similarly to STN, a solution to a TSTN is an assignment of all time-points which satisfies all t-simple temporal constraints, and a TSTN is said to be consistent iff it has at least one solution. Similarly to STN again, a TSTN can be represented graphically by its *distance graph*, which contains one arc  $y \rightarrow x$  labeled by  $-dmin(x, y)$  for each temporal constraint  $y - x \geq dmin(x, y)$ . The only difference with the standard distance graph associated with an STN is that arc labels are not necessarily constant in the case of TSTN.

A relevant notion associated with a t-simple temporal constraint  $y - x \geq dmin(x, y)$  is its *delay function*, defined by  $delay(x, y) = x + dmin(x, y) - y$ . Informally,  $delay(x, y)$  is the delay obtained in  $y$  if a transition in minimum time from  $x$  to  $y$  is triggered at time  $x$ . This delay corresponds to the difference between the minimum arrival time associated with the transition ( $x + dmin(x, y)$ ) and the required arrival time ( $y$ ). A strictly negative delay corresponds to a transition ending before deadline  $y$ . A strictly positive delay corresponds to a violation of the temporal constraint. A null delay corresponds to an arrival right on time.

### Properties of TSTNs

One fundamental question is whether standard properties that hold for STN can be generalized to TSTN. The answer is that some of the STN results can be extended to TSTN, whereas others cannot.

#### Local Consistency of a T-simple Temporal Constraint

The first step is to define how to handle each t-simple temporal constraint, and more precisely to define how each constraint can be used to prune the earliest and latest times associated with each time-point.

In STN, a simple temporal constraint  $y - x \geq c$  is *propagated* following the domain reduction rules given below:

$$\mathbf{d}(y) \leftarrow \mathbf{d}(y) \cap [\min(\mathbf{d}(x)) + c, +\infty[ \quad (1)$$

$$\mathbf{d}(x) \leftarrow \mathbf{d}(x) \cap ] - \infty, \max(\mathbf{d}(y)) - c] \quad (2)$$

In TSTN, a t-simple temporal constraint  $y - x \geq dmin(x, y)$  is propagated using more general rules, which are written as follows:

$$\mathbf{d}(y) \leftarrow \mathbf{d}(y) \cap [earr(\min(\mathbf{d}(x))), +\infty[ \quad (3)$$

$$\mathbf{d}(x) \leftarrow \mathbf{d}(x) \cap ] - \infty, ldep(\max(\mathbf{d}(y)))] \quad (4)$$

with  $earr$  and  $ldep$  two functions defined by  $earr(a) = \min\{+\infty\} \cup \{b \in \mathbf{d}(y) \mid b - a \geq dmin(a, b)\}$  and  $ldep(b) = \max\{-\infty\} \cup \{a \in \mathbf{d}(x) \mid b - a \geq dmin(a, b)\}$  respectively. Rule 3 updates the earliest time associated with  $y$ , while Rule 4 updates the latest time associated with  $x$ . Informally, if  $x$  and  $y$  are seen as the start and end times of a transition, then Rule 3 prunes the domain of  $y$  by computing the earliest arrival time in  $y$  such that the temporal constraint is satisfied when starting the transition as soon as possible (at time  $\min(\mathbf{d}(x))$ ). Rule 4 prunes the domain of  $x$  by computing the latest departure time from  $x$  such that the temporal constraint is satisfied when arriving in  $y$  as late as possible (at time  $\max(\mathbf{d}(y))$ ).

Two main results concerning the propagation rules can be established:

1. Rules 3 and 4 establish *bound arc-consistency*, which means that after the application of the rules, the time bounds associated with  $x$  and  $y$  participate to an assignment which satisfies the temporal constraint.
2. For t-simple temporal constraints which have a specific monotonicity property called *delay-monotonicity* (which holds for STN), any value remaining in the domain of  $x$  (resp.  $y$ ) after the application of the pruning rules has a support in the domain of  $y$  (resp.  $x$ ). When delay-monotonicity is violated, the latter property does not necessarily hold. However, delay-monotonicity is a very natural property, which expresses that the sooner an activity starts, the sooner it ends.

Concerning the way  $earr$  and  $ldep$  are computed in practice, several approaches can be considered. For simple temporal constraints  $y - x \geq c$ , an analytic formulation is possible (see Rules 1 and 2). Analytic formulations can also be derived for several standard temporal constraints used in time-dependent scheduling. In the general case, an iterative approximated method based on linear interpolation techniques can be used.

**Global Consistency of a TSTN** When considering TSTNs, which involve sets of t-simple temporal constraints, three main results can be established:

1. Similarly to STN, if all constraints of a TSTN are made bound arc-consistent using Rules 3-4, then the schedule which assigns to each time-point its earliest (resp. latest) possible time is a solution of the TSTN.
2. Contrarily to STN, even if all constraints of a TSTN are made bound arc-consistent using Rules 3-4, non-extremal values in the variable domains are not necessarily globally consistent, which means that these values may not participate to any solution of the TSTN. This result holds even if the delay function has the monotonicity property mentioned before.
3. Assume that the delay-monotonicity property holds and that cycles of the distance graph involve only simple temporal constraints. Then, every value remaining in the variable domains is globally consistent (sufficient condition for making the STN result applicable again).

### Solving TSTN

Algorithms can be introduced to determine the consistency of a TSTN and to compute the earliest and latest possible times associated with each time-point. We consider a context in which temporal constraints may be added or removed from the problem, which happens when activities are added or removed during the search for good plans or schedules.

**Constraint Propagation** Similarly to existing work on STN (Cesta and Oddi 1996; Gerevini, Perini, and Ricci 1996; Shu, Effinger, and Williams 2005), we use constraint propagation techniques for computing min and max bounds

of time-points. These techniques apply Rules 3 and 4 for revising constraints until a fixed point is reached. We use a constraint propagation scheme in which we maintain two lists containing constraints to be revised: one list for propagating constraints modifying min bounds (application of Rule 3), and one list for propagating constraints modifying max bounds (application of Rule 4).

**Negative Cycle Detection** In STN, inconsistency is equivalent to the existence of a negative cycle in the distance graph. Efficient approaches have been proposed on STN for detecting such negative cycles (Cervoni, Cesta, and Oddi 1994; Cesta and Oddi 1996). These approaches are able to detect negative cycles quite fast by maintaining so-called *propagation chains*. The latter can be seen as explanations for the current min and max bounds of the different time-points. The intuition is that if a propagation cycle  $x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_n \rightarrow x_1$  is detected for min bounds, then this means that the min value of  $x_1$  modified the min value of  $x_2$ , which modified the min value of  $x_n$  which modified the min value of  $x_1$ . By traversing this propagation cycle a sufficient number of times, the domain of  $x_1$  can be entirely pruned. A similar result holds for max bounds.

In TSTN, the relationship between TSTN inconsistency and the existence of a negative cycle is a bit more complex:

1. For TSTN in general, the existence of a propagation cycle does not necessarily imply inconsistency. One reason is that in TSTN, domain reductions obtained by traversing cycles again and again may become smaller and smaller due to the time-dependent aspect, and consequently domain reductions obtained may not necessarily prune all values in the domains.
2. Strict monotonicity of the delay function does not suffice for establishing the equivalence again.
3. If all constraints  $y - x \geq dmin(x, y)$  involved in a propagation cycle for min bounds (resp. max bounds) are delay-monotonic and if they all have a non-decreasing duration  $dmin(x, y)$  (resp. a non-increasing duration) with regard to their two arguments  $x$  and  $y$ , then the TSTN is inconsistent. Such a condition guarantees that a propagation cycle does not become “less negative” when traversed again and again.
4. For a TSTN such that all cycles of the distance graph involve only simple temporal constraints (and not t-simple ones), the existence of a propagation cycle implies inconsistency.

In all applications which we have treated so far with TSTN, one of the sufficient conditions mentioned above held. If the sufficient conditions are not satisfied, several options can be considered, leading to different trade-offs between exact inconsistency detection and number of constraint revisions performed.

**Complexity** Let  $V$  be the set of time-points in the problem and let  $C$  be the set of temporal constraints. If the existence of a propagation cycle implies inconsistency, then it

is possible to establish bound arc-consistency in  $O(|V||C|)$  constraint revisions, with an algorithm which uses Rules 3-4 for revising constraints, a FIFO ordering of the propagation queue, and a propagation cycle detection technique. As in STNs, the bound obtained does not depend on the size of the variable domains. Note however that in TSTN, even if the number of constraint revisions is polynomial, the time required to revise one constraint is not necessarily  $O(1)$ .

**Constraint Depropagation** Constraint propagation techniques are directly able to handle constraint addition or constraint strengthening. As for constraint removal or constraint weakening, constraint depropagation strategies defined in (Shu, Effinger, and Williams 2005) for STN can be directly reused for TSTN. These strategies allow min and max bounds of temporal variables to be recomputed at minimum cost. They avoid reinitializing all variable domains and repropagating all constraints from scratch when a constraint is removed or weakened. The basic idea is to use propagation chains to determine which variable domains must be reinitialized and which constraints need to be revised.

**Constraint Revision Ordering** A last technique is used for minimizing the number of constraint revisions. This can be particularly useful for TSTN, for which revising one constraint can be significantly more costly than in STN. The proposed approach extends a technique developed for STN<sup>-</sup> (Gerevini, Perini, and Ricci 1996), a subclass of STN in which every constraint must be rewritable as  $y - x \geq c$  with  $c \geq 0$ . The idea consists in building the strongly connected components (SCCs) of the distance graph, and in propagating constraints following a topological order of SCCs. Such a topological order can be maintained upon additions and removals of constraints based on dynamic graph algorithms (Haeupler et al. 2012; Roditty and Zwick 2008).

**Management of Unary Constraints** In the constraint propagation scheme used, unary constraints  $x \leq a$  and  $x \geq b$  are actually revised first because their revision is easy. Another argument is that if such constraints were handled as distance constraints with regard to a reference temporal position  $x_0$  (constraints  $x - x_0 \leq a$  and  $x - x_0 \geq b$ ), then the distance graph would contain several additional arcs which could interfere with the definition of SCCs and degrade the efficiency of constraint propagation.

## Experiments

All techniques previously introduced (constraint propagation, propagation cycle detection, constraint depropagation, SCCs ordering, specific management of unary constraints) are integrated in the InCELL solver (Pralet and Verfaillie 2013a), which allows TSTN to be represented and manipulated. Experiments were realized on TSTNs modeling temporal problems associated with the management of so-called agile satellites, which have the capacity to move around their center of gravity to point to a particular direction. Results showed that using TSTN for managing temporal constraints

is feasible in terms of computation times. They also showed that modeling time-dependent constraints can lead to better schedules (better in terms of both temporal flexibility and number of achievable tasks) than an approach which transforms TSTN to STN by modeling t-simple temporal constraints  $y - x \geq dmin(x, y)$  as simple temporal constraints  $y - x \geq c$  with  $c$  the maximum of all possible minimal distances to be respected between  $x$  and  $y$ , defined by  $c = \max_{a \in \mathbf{d}(x), b \in \mathbf{d}(y)} dmin(a, b)$ .

### Perspectives

Two main perspectives can be provided. First, it would be interesting to combine TSTN with optimization, in order to handle problems in which a different reward is obtained depending on the time at which an activity is performed. On this point, it would be interesting to try and reuse works combining STN and linear objective functions (Morris et al. 2004). Second, TSTN could be extended to problems in which there is an uncertainty about the duration of tasks. On this point, existing work on STN with uncertainties (STNU (Vidal and Fargier 1999)) should be considered.

### References

- Cervoni, R.; Cesta, A.; and Oddi, A. 1994. Managing dynamic temporal constraint networks. In *Proc. of AIPS-94*, 13–18.
- Cesta, A., and Oddi, A. 1996. Gaining efficiency and flexibility in the simple temporal problem. In *Proc. of TIME-96*, 45–50.
- Cheng, T.; Ding, Q.; and Lin, B. 2004. A concise survey of scheduling with time-dependent processing times. *European Journal of Operational Research* 152:1–13.
- Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49:61–95.
- Gawiejnowicz, S. 2008. *Time-dependent scheduling*. Springer.
- Gerevini, A.; Perini, A.; and Ricci, F. 1996. Incremental algorithms for managing temporal constraints. In *Proc. of ICTAI-96*, 360–365.
- Haeupler, B.; Kavitha, T.; Mathew, R.; Sen, S.; and Tarjan, R. 2012. Incremental cycle detection, topological ordering, and strong component maintenance. *ACM Transactions on Algorithms* 8(1).
- Morris, P.; Morris, R.; Khatib, L.; Ramakrishnan, S.; and Bachmann, A. 2004. Strategies for global optimization of temporal preferences. In *Proc. of CP-04*, 408–422.
- Pralet, C., and Verfaillie, G. 2013a. Dynamic online planning and scheduling using a static invariant-based evaluation model. In *Proc. of ICAPS-13*.
- Pralet, C., and Verfaillie, G. 2013b. Time-dependent simple temporal networks: Properties and algorithms. *RAIRO Operations Research* 47(2):173–198.
- Roditty, L., and Zwick, U. 2008. Improved dynamic reachability algorithms for directed graphs. *SIAM Journal on Computing* 37(5):1455–1471.

- Shu, I.; Effinger, R.; and Williams, B. 2005. Enabling fast flexible planning through incremental temporal reasoning with conflict extraction. In *Proc. of ICAPS-05*, 252–261.
- Stergiou, K., and Koubarakis, M. 2000. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence* 120:81–117.
- 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.