Research Project Proposal on FUR 2019

Title: Extending Uncertainty in Temporal Constraint Networks (EUTCN)

Total duration of the AdR: 12

Total cost of the AdR: 25000€

Number of months that will be co-financed and project sources (if any): 6 using Roberto Posenato FUR and CCIA funds.

Abstract:

Temporal constraint networks have been adopted in different research areas for reasoning on temporal requirements like planning and scheduling, business process and healthcare informatics. In such areas the proposed temporal constraint models allow the representation of both conditional and contingent constraints. In general, *conditional constraints* hold only when some specific situations happen. Thus, it is possible to specify in a compact way constraints holding in different contexts. Usually, such contexts are determined at run-time since related to external events. *Contingent constraints* allow the representation of event occurrences that are not under the control of the executing agent but are tied to happen within some specified time interval. Such a time interval cannot be modified by the executing agent. Therefore, it is necessary to guarantee that all the other constraints are consistent for all possible event occurrences.

Since 2009, the project proposer (R.P.), with others, contributed to formally define Conditional Simple Temporal Network with Uncertainty model (CSTNU) to represent such information and recently contributed to the proposal of a sound-and-complete algorithm for checking the dynamic controllability (DC) of CSTNUs. (A CSTNU is DC if it is possible to execute it satisfying all constraints no matter which combination of contexts and contingent durations occurs.) He also contributed to extend the theoretical foundations of related classes of temporal networks like, for example, the Conditional Simple Temporal Network with Partial Shrinkable Uncertainty model (CSTNPSU). In CSTNPSU, each contingent constrain is represented by a designer-wanted duration range that can be partially shrunk at *run-time* to guarantee the DC of the network. For CSTNPSU, R.P., with others from Ulm University (Germany), proposed a sound DC checking algorithm.

The main research objectives of this project are 1) to develop new faster algorithms for checking CSTNUs and 2) to strengthen the DC checking techniques for CSTNPSU to determine a sound-and-complete DC checking algorithm.

This document is organized as follows. In 'Extended Synopsis' section, I present a background on Temporal Constraint Network; in 'Expected Impact' section, I present the research objectives and their possible impact on some research areas; in 'Feasibility' section, I present a temporal planning of the activities to show its feasibility and possible risks of this project.

Department research area (main topic – cancel those irrelevant): Computing

Department research sub-area (at most one sub-topic): Computation theory

Keywords: (free keywords) Artificial intelligence, intelligent systems, temporal reasoning, constraint networks, dynamic controllability

Extended Synopsis: (max 10.000 chars including references)

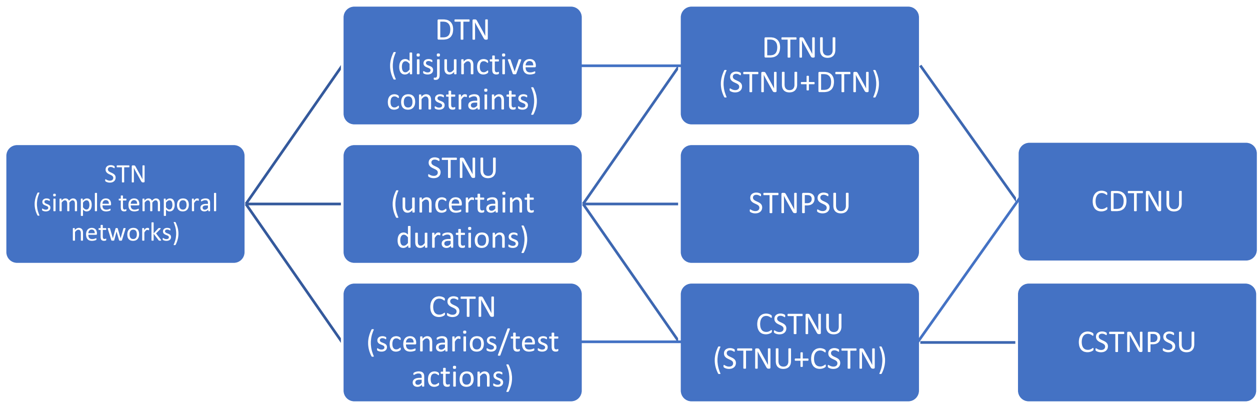


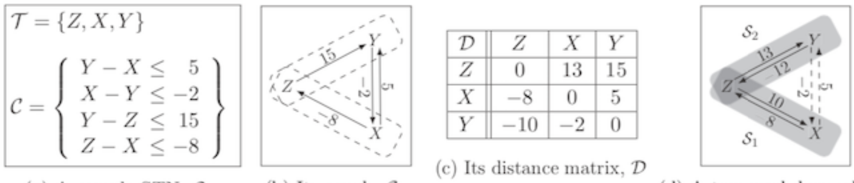
Figure 1: A hierarchy of temporal networks

Temporal Networks

This section provides necessary background on temporal networks.
A Simple Temporal Network (STN) is a set of real-valued variables, called time-points, together with a set of temporal constraints on those time-points [5].
Although an STN can represent such common kinds of constraints as release times, deadlines, inter-action constraints, and constraints on action durations, it is not able to represent other common kinds of features. As a result, over the years, different kinds of temporal networks have been introduced that augment STNs to include new features.
For example, as illustrated in Fig. 1, a Simple Temporal Network with Uncertainty (STNU) augments an STN to include contingent links that can be used to represent actions with uncertain durations [15]; a Conditional Simple Temporal Network (CSTN) augments an STN to include observation time-points and conditional constraints that can be used to represent test actions that generate information [19]; and a Conditional Simple Temporal Network with Uncertainty (CSTNU) combines the features of STNUs and CSTNs [9]. STNPSU introduces flexibility to contingent links, while CSTNPSU introduces the same kind of flexibility to CSTNUs [4,11,18].
An important problem for any kind of temporal network is that of determining whether it is consistent or, for more complex networks, dynamically controllable (DC). For example, a CSTN is DC if there is a dynamic strategy for executing its time-points that can guarantee that all relevant constraints will be satisfied no matter which test outcomes are observed in real-time.
The following sections introduce relevant background information for some of the temporal networks in Fig. 1, and summarize the progress that has been made in developing both the theoretical foundations of and algorithms for managing these kinds of networks.

Simple Temporal Networks

A Simple Temporal Network (STN) is a data structure for representing and reasoning about time. Formally, an STN is a pair S=(T, C), where T is a set of real-valued variables, called time-points, and C is a set of binary difference constraints each having the form, $Y-X\leq\delta$, for some $X, Y\in T$, and $\delta\in\mathbb{R}$ [5]. For example, if X and Y are time-points representing the starting and ending times of some action, then the constraint, $Y-X\leq 5$, would stipulate that the duration of that action, $Y-X$, must be no more than 5.



(a) A sample STN, $\mathcal{S}$ (b) Its graph, $\mathcal{G}$ (c) Its distance matrix, $\mathcal{D}$ (d) A temporal decoupling

Figure 2: A sample STN, its graph, its distance matrix, and a sample temporal decoupling

A *solution* for an STN is a set of assignments to its time-points that satisfies all of its constraints. An STN is called *consistent* if it has a solution. For example, {Z=0, X=9, Y=12} is a solution for the STN in Fig. 2a; thus, that STN is consistent.

Each STN S=(T,C) has an associated graph G, where the time-points in T correspond one-to-one to the nodes in G, and the binary constraints in C correspond one-to-one to the labeled, directed edges in G. In particular, each constraint $Y-X \leq \delta$ corresponds to an edge from X to Y labeled by δ in G. The graph for the sample STN is shown in Fig. 2b

Paths in an STN graph correspond to implicit constraints that must be satisfied by any solution for that STN, and shortest paths correspond to strongest implicit constraints. Thus, the all-pairs, shortest-paths (APSP) matrix D for G plays an important role for STNs. D is called the distance matrix for S (and G). The Fundamental Theorem of STNs stipulates that the following are equivalent:

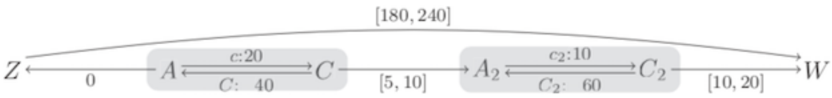
1) the STN S is consistent;
2) its graph G has no negative loops; and
3) its distance matrix D has non-negative entries down its main diagonal [5,6].

In view of this result, the STN Consistency Problem (n time-points and m constraints) can be solved in *O(mn)* time (e.g., by the Bellman-Ford algorithm), and the distance matrix D can be computed in $O(n^3)$ time (e.g., by the Floyd-Warshall algorithm).

Simple Temporal Networks with Uncertainty

Although STNs can represent a wide variety of temporal constraints, they are not able to represent actions with uncertain durations. Since such actions are common in many domains (e.g., in developing controllers for autonomous spacecraft [50]), researchers defined a Simple Temporal Network with Uncertainty (STNU) to include *contingent links* [21,20,15]. Each *contingent link* has the form (A,x,y,C), where A is called the activation time-point, and C is called the contingent time-point. Typically, the agent executing the network only controls the execution of A. Once A executes, the environment (which may be viewed as an adversarial player) determines when C executes, subject only to the constraint that $C-A \in [x, y]$. For example, a taxi ride to the airport that is known to take between 20 and 40 minutes might be represented by the contingent link (A,20,40,C), where A represents the starting time of the taxi ride, and C represents its ending time. The passenger controls when he/she gets into the taxi, but only passively observes the actual duration of the ride once the taxi arrives at its destination.

Each STNU has an associated graph, which is similar to an STN graph, except that each contingent link, (A,x,y,C), is represented by a pair of labeled edges: a *lower-case edge* from A to C, labeled by *c:x*, and an *upper-case edge* from C to A, labeled by *C:−y*. The lower-case edge represents the uncontrollable possibility that the contingent duration might take on its minimum value x; the upper-case edge represents the uncontrollable possibility that the contingent duration might take on its maximum value y. An STNU graph for a taxi ride to the airport is shown in Fig. 3.



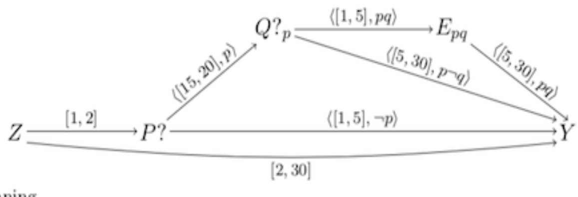
Time-Point	Meaning
Z	Begin ($Z = 0$ corresponds to 12:00 Noon)
A	Get into taxi
C	Exit taxi at airport (between 20 and 40 minutes after A)
A_2	Enter line for security check
C_2	Exit security check (between 10 and 60 minutes after A_2)
W	Arrive at gate—hopefully, between 3:00 and 4:00 p.m. (i.e., $W \in [180, 240]$)

Figure 3: An STNU graph for a taxi ride to the airport (contingent links shaded)

The most important property of an STNU is whether it is *dynamically controllable (DC)* [13]. An STNU is DC if there exists a dynamic strategy for executing its non-contingent time-points that guarantees that all constraints in the network will be satisfied no matter how the durations of the contingent links turn out—within their specified bounds. Crucially, a dynamic strategy can react, in real-time, to the executions of contingent time-points; however, its execution decisions are only allowed to depend on past events, not advance knowledge of future events.
Much of the research on STNUs has been devoted to finding efficient DC-checking algorithms for STNUs. The most recent result propose an $O(n^3)$-time algorithm in 2014 [12,16,1].

Conditional Simple Temporal Networks

Neither STNs nor STNUs can accommodate test actions that generate new information during execution (e.g., a blood test). To meet this need, Tsamardinos et al. [19] defined a Conditional Simple Temporal Network (CSTN) that augments an STN—not an STNU—to include *observation time-points*. Each observation time-point has a corresponding propositional letter; when the observation time-point is executed, its corresponding propositional letter receives a boolean value—true or false. Crucially, the agent executing the network does not determine that value; instead, it is determined by the environment, which can be viewed as a (possibly adversarial) player. For example, P? might represent the time that a patient's blood pressure is measured, and p=true might represent that the patient has high blood pressure. Each time-point/constraint in a CSTN can have a propositional label (i.e., a conjunction of positive or negative literals) that specifies the ***scenarios*** in which it is relevant. For example, a time-point X (or constraint $Y-X\leq 7$) labeled by p¬q would only be executed (must be satisfied) in scenarios where p=true and q=false. Each CSTN has an associated graph, where each time-point is represented by a node in the graph, and each labeled constraint, $(Y-X\leq\delta, \alpha)$, is represented by an edge from X to Y, annotated by the labeled literals. For example, Fig. 4 shows a CSTN graph for a hypothetical medical procedure in which emergency treatment is performed only if a blood test twice returns a positive result.



Time-Point	Meaning
Z	Begin
$P?$	Do Blood Test (generates truth value for p; $p = true$ indicates positive reaction)
$Q?_p$	Repeat Test (applicable if $p = true$; generates value for q; $q = true$ indicates positive reaction)
E_{pq}	Do Emergency Treatment (only applicable if $p = q = true$)
Y	End

Figure 4: A CSTN

A CSTN is called *dynamically consistent (DC)* if there exists a dynamic strategy for executing its time-points such that all relevant constraints are guaranteed to be satisfied no matter which scenario is incrementally revealed during execution (i.e., as the observation time-points execute). R.P., with others, made the following significant contributions to research on CSTNs:

1) Presented two sound-and-complete DC-checking algorithms for CSTNs, one based on the propagation of labeled constraints, the other on Timed Game Automata (TGAs) and Monte Carlo Tree Search, and demonstrated its practicality across a variety of networks [10,17,2].
2) Gave a method for constructing the earliest-first execution strategy for DC CSTNs [10,7].

Conditional Simple Temporal Networks with Uncertainty

Because many applications involve both actions with uncertain durations and test actions that generate new information, R.P., with others, introduced Conditional Simple Temporal Networks with Uncertainty (CSTNUs) which augment STNs to include both the contingent links from STNUs and the observation time-points and conditional constraints from CSTNs [9]. They defined a dynamic controllability property for CSTNUs that generalizes the corresponding properties for STNUs and CSTNs. Their initial work generated a variety of sound constraint-propagation rules [2,3]. More recently, in 2018, they presented two sound-and-complete DC-checking algorithms for CSTNUs: one that propagates labeled constraints directly in the input CSTNU graph, and one that reduces the DC-checking problem for CSTNUs to the DC-checking problem for CSTNs [8].

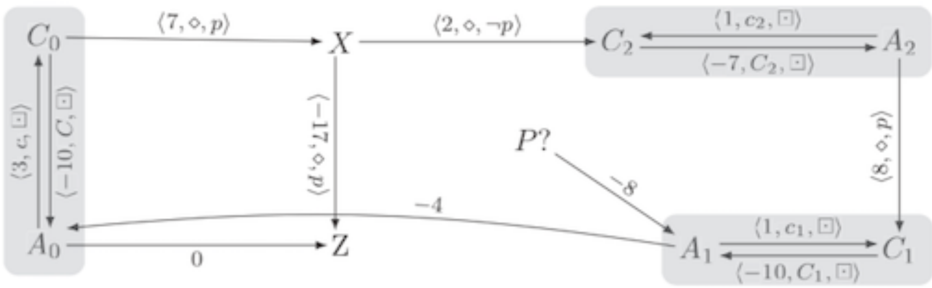


Figure 5: A CSTNU graph with 3 contingent links and one observation time-point $P?$

Fig. 5 shows a sample CSTNU graph that contains three contingent links (shaded) and one observation time-point, P?. Although it is far from obvious, the DC-checking algorithms confirm that the CSTNU shown in Fig. 5 is not DC.

On the experimental evaluation front, R.P. implemented all the algorithms for CSTN and CSTNU DC checking in Java and made them freely available inside a new tool--temporal network editor—called CSTNU Editor [17]. Fig. 6 shows a screenshot of CSTN Editor.

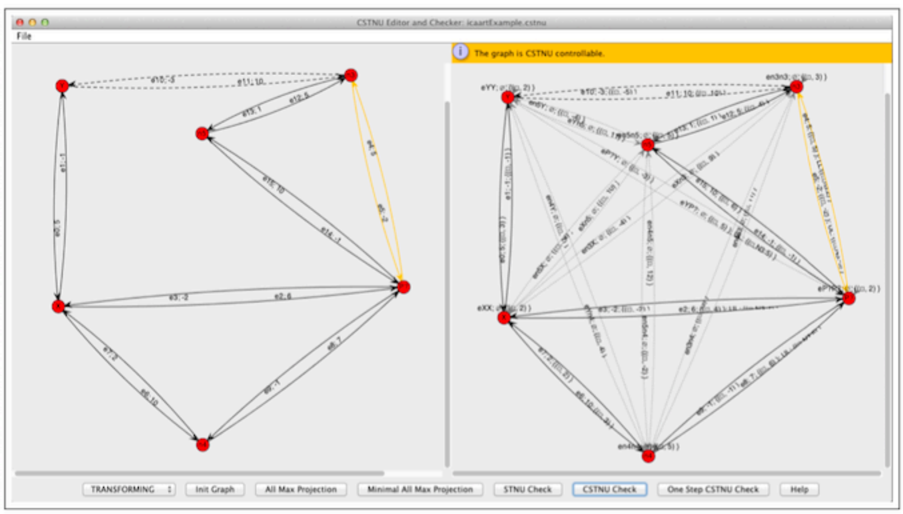


Figure 6: A screen shot of the CSTNU editor and checker.

(Conditional) Simple Temporal Networks with Partial Shrinkable Uncertainty

In some contexts [11,18], a contingent link (A,l,u,C) represents a hard constraint to satisfy and it is required a greater flexibility in its management. R.P., with others, introduced a new version of contingent link concept introducing *guarded links.* A guarded link (A,[x,x'][y',y]C) represents a contingent link where the lower bound can be chosen in the range [x,x'] and the upper bound in the range [y',y] [11,18]. In this way, it is possible to have a flexibility in the definition of contingent links where the optimal range [x,y] can be reduced if it is necessary but with the guarantee that it cannot be reduced to a sub range of [x',y']. They also proposed sound algorithms for checking the DC in CSTNs having guarded links [18] and in CSTNUs [4].

References

1. M. Cairo, L. Hunsberger, and R. Rizzi. Faster dynamic controllability checking for simple temporal networks with uncertainty. In 25th Int. Symp. on Temporal Repr. and Reasoning (TIME 2018), vol 120, pages 8:1–8:16, 2018.
2. C. Combi, L. Hunsberger, and R. Posenato. An algorithm for checking the dynamic controllability of a conditional simple temporal network with uncertainty. In Proc. 5th Int. Conf. on Agents and Artificial Intelligent (ICAART 2013), vol 2, pages 144–156, 2013.
3. C. Combi, L. Hunsberger, and R. Posenato. An algorithm for checking the dynamic controllability of a conditional simple temporal network with uncertainty - revisited. In Agents and Artificial Intelligence, vol 449 of Communications in Computer and Information Science (CCIS), pages 314–331. 2014.
4. C. Combi and R. Posenato, "Extending Conditional Simple Temporal Networks with Partially Shrinkable Uncertainty," in *25th International Symposium on Temporal Representation and Reasoning (TIME 2018)*, vol. 120 of *LIPIcs*, pp. 9:1–9:15, doi:10.4230/LIPIcs.TIME.2018.9.
5. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. Artificial Intelligence, 49(1-3):61–95, 1991.
6. L. Hunsberger. Efficient execution of dynamically controllable simple temporal networks with uncertainty. Acta Informatica, 53(2):89–147, 2015.
7. L. Hunsberger and R. Posenato. Simpler and faster algorithm for checking the dynamic consistency of conditional simple temporal networks. In Proc. 26th Int. Joint Conf. on Artificial Intelligence, (IJCAI), pages 1324–1330, 2018.
8. L. Hunsberger and R. Posenato. Sound-and-Complete Algorithms for Checking the Dynamic Controllability of Conditional Simple Temporal Networks with Uncertainty. In 25th Int. Symp. on Temporal Representation and Reasoning (TIME 2018), vol 120 of LIPIcs, pages 14:1–14:17, 2018.
9. L. Hunsberger, R. Posenato, and C. Combi. The Dynamic Controllability of Conditional STNs with Uncertainty. In PlanEx at ICAPS 2012, pages 1–8, 2012.
10. L. Hunsberger, R. Posenato, and C. Combi. A sound-and-complete propagation-based algorithm for checking the dynamic consistency of conditional simple temporal networks. In 22nd Int. Symp. on Temporal Representation and Reasoning (TIME 2015), pages 4–18, 2015.
11. A. Lanz, R. Posenato, C. Combi, and M. Reichert. Controllability of time-aware processes at run time. In Proc. On the Move to Meaningful Internet Systems: OTM 2013 Conf.s, vol 8185 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2013.
12. P. Morris. Dynamic controllability and dispatchability relationships. In Integration of AI and OR Techniques in Constraint Programming, vol 8451 of Lecture Notes in Computer Science (LNCS), pages 464–479. Springer, 2014.
13. P. Morris. The mathematics of dispatchability revisited. In 26th Int. Conf. on Automated Planning and Scheduling (ICAPS-2016), pages 244–252, 2016.
14. P. H. Morris and N. Muscettola. Managing temporal uncertainty through waypoint controllability. In Proc. of the 7th Int. Joint Conf. on Artificial Intelligence (IJCAI-99), 1999.
15. P. H. Morris, N. Muscettola, and T. Vidal. Dynamic control of plans with temporal uncertainty. In IJCAI 2001, pages 494–502, 2001.
16. M. Nilsson, J. Kvarnstrom, and P. Doherty. Incremental dynamic controllability in cubic worstcase time. In 21st Int. Symp. on Temporal Representation and Reasoning (TIME-2014), 2014.
17. R. Posenato, "The CSTNU Toolset." http://profs.scienze.univr.it/~posenato/software/cstnu/ 2019.
18. R. Posenato, A. Lanz, C. Combi, and M. Reichert, "Managing time-awareness in modularized processes," Software & Systems Modeling, vol. 18, no. 2, pp. 1135–1154, Apr. 2019.
19. I. Tsamardinos, T. Vidal, and M. E. Pollack. CTP: A new constraint-based formalism for conditional, temporal planning. Constraints, 8:365–388, 2003.
20. T. Vidal and Hélène Fargier. Contingent durations in temporal CSPS: from consistency to controllabilities. In Fourth Int. Workshop on Temporal Representation and Reasoning (TIME-97), pages 78–85, 1997.
21. T. Vidal and Malik Ghallab. Temporal constraints in planning: Free or not free? CONSTRAINT, 1995.

22. S Cheikhrouhou, S Kallel, N Guermouche, and M Jmaiel. A Survey on Time-aware Business Process Modeling. Technical report, Centre pour la Communication Scientifique Directe, 2013.
23. J. Eder, E. Panagos, and M. Rabinovich, "Time Constraints in Workflow Systems," in Advanced Information Systems Engineering (CAiSE 1999), 1999, vol. 1626, pp. 286–300.
24. R. Lenz and M. Reichert, "IT Support for Healthcare Processes," in BPM, 2005, pp. 354–363.
25. R. Posenato, A. Lanz, C. Combi, and M. Reichert, "Managing time-awareness in modularized processes," *Software & Systems Modeling*, vol. 18, pp. 1135–1154, Apr. 2019. doi:10.1007/s10270-017-0643-4.

Expected Impact: (max 2.000 chars)

The determination of faster DC checking algorithms for CSTN/CSTNU/CSTNPSU networks is crucial to allow a greater applicability of such models in different research areas like, for example, the Workflow Management Systems or Process Aware Information System (PAIS) [22,23,24].
Indeed, in a workflow system, the management of temporal aspects, like deadlines, temporal distance between activities, minimal/maximum durations of tasks, is becoming an important feature for allowing a better representation and management of real business plans [11,22,23,24].
In [11,25], R.P., with others, shown how it is possible to represent and manage temporal aspects of workflow schemata using temporal networks like STNU/CSTNU. An important aspect for a successful application of temporal networks in PAIS is to guarantee that it is possible to check networks in a small amount of time for making the designer task easier.
Although existing DC-checking algorithms for CSTNs and CSTNUs are sound and complete and guaranteed to terminate, anecdotal evidence suggests that adding more constraint-propagation rules can lead to faster convergence, and hence faster algorithms. Therefore, this project will explore the use of expanded sets of propagation rules for CSTNUs, empirically evaluating the performance of the resulting algorithms to determine which rule-sets provide the best performance. Although modifying existing algorithms can lead to better performance, it may also be possible to generate entirely new sets of propagation rules that lead to faster algorithms [1]. Therefore, this project will also search for alternative sets of constraint-propagation rules for CSTNUs that may lead to faster DC-checking algorithms.
As regards STNPSU/CSTNPSU, current DC checking algorithm have been proved to be only sound but not complete. Therefore, it is necessary to improve current algorithms adding the necessary features that allow them to be complete.

Feasibility: (max 2.000 chars)

1. Technical capability and temporal planning
    1. The project provides to make a performance analysis of current DC checking algorithms and to evaluate some expanded sets of propagation rules for CSTNUs for a possible improvement. All considered algorithms have been already implemented by R.P. in Java as well as some expanded sets of propagation rules. Such software suite can be considered a reference implementation that must be analyzed and optimize. Therefore, such task can be easily conduct by a post-doc (supervised by R.P.) having a sufficient experience in Java programming in 2 man-months (MMs).
    2. The project provides to search for alternative sets of constraint-propagation rules for CSTNUs that may lead to faster DC-checking algorithms. There is already a paper offering an alternative sets of constraint-propagation rules for STNU. It is necessary to study how to extend such sets considering also scenarios, for which the already defined techniques used for CSTNs may be sufficient. The resulting algorithm, then, must be implemented and its practical performances evaluated with respect to the performances of algorithm considered in item 1.1. A suite of benchmarks have been already published by R.P. [17]. The expected MMs are 6.

   3. The project provides to improve DC checking algorithms for STNPSU/CSTNPSU making them complete. Some preliminary work have been already done by R.P. and Andrea Lanz. It is necessary to revise and complete such a preliminary work under the supervision of R.P. The expected MMs are 4.

2. Risk
   The main risk of such a project is represented by the goal 1.2 for which it is possible that we are not able to find an alternative set of rules for CSTNU/CSTNPSU that allows the determination of a better DC-checking algorithm based on constraint-propagations. In such a case, we will show that the extension of current alternative known set of rules are not suitable for CSTNU/CSTNPSU.

UNIVERSITÀ di VERONA | Dipartimento di INFORMATICA

Short CV of the proposer: (max 5.000 chars)

Roberto Posenato took a degree in Computer Science in 1991 and a doctor's degree in Computational Mathematics in 1996 at the University of Milan (Italy).
Currently, he is an associate professor at the Department of Computer Science, University of Verona (Italy).
He has been a lecturer for some courses in the theory of algorithms, computational complexity and databases since 1996.
The main research interests are related to resolution algorithms for combinatorial optimization problems, with emphasis on temporal constraint networks.
He is also interested in the study of time reasoning in workflow systems and its possible applications.
He has two main international collaborations: the University of Ulm (Germany), the University of Klagenfurt (Austria) and Vassar College (USA).
He is a reviewer for national and international journals and conferences.
Moreover, he has been involved in several national/international research projects.

Up to 10 of most relevant publications of the proposer

1. C. Combi, R. Posenato, L. Viganò, and M. Zavatteri, "Conditional Simple Temporal Networks with Uncertainty and Resources," *The Journal of Artificial Intelligence Research*, vol. 64, pp. 931–985, 2019. doi:10.1613/jair.1.11453.
2. R. Posenato, A. Lanz, C. Combi, and M. Reichert, "Managing time-awareness in modularized processes," *Software & Systems Modeling*, vol. 18, pp. 1135–1154, Apr. 2019. doi:10.1007/s10270-017-0643-4.
3. L. Hunsberger and R. Posenato, "Sound-and-Complete Algorithms for Checking the Dynamic Controllability of Conditional Simple Temporal Networks with Uncertainty," in *25th International Symposium on Temporal Representation and Reasoning (TIME 2018)* (N. Alechina, K. Nørvåg, and W. Penczek, eds.), vol. 120 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 14:1–14:17, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.TIME.2018.14.
4. R. Posenato, F. Zerbato, and C. Combi, "Managing decision tasks and events in time-aware business process models," in *Business Process Management - 16th International Conference (BPM 2018)* (M. Weske, M. Montali, I. Weber, and J. vom Brocke, eds.), vol. 11080 of *Lecture Notes in Computer Science (LNCS)*, pp. 102–118, Springer, 2018. doi:10.1007/978-3-319-98648-7_7.
5. L. Hunsberger and R. Posenato, "Simpler and faster algorithm for checking the dynamic consistency of conditional simple temporal networks," in *Proceedings of the 26th Inter- national Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 1324–1330, International Joint Conferences on Artificial Intelligence Organization, July 2018. doi:10.24963/ijcai.2018/184.
6. C. Comin, R. Posenato, and R. Rizzi, "Hyper temporal networks," *Constraints*, vol. 22, pp. 152– 190, Apr. 2017. doi:10.1007/s10601-016-9243-0.
7. A. Cimatti, L. Hunsberger, A. Micheli, R. Posenato, and M. Roveri, "Dynamic controllability via Timed Game Automata," *Acta Informatica*, vol. 53, pp. 681–722, Oct. 2016. doi:10. 1007/s00236-016-0257-2.
8. L. Hunsberger, R. Posenato, and C. Combi, "A sound-and-complete propagation-based algorithm for checking the dynamic consistency of conditional simple temporal networks," in *22nd International Symposium on Temporal Representation and Reasoning (TIME 2015)* (F. Grandi, M. Lange, and A. Lomuscio, eds.), pp. 4–18, IEEE CPS, Sept. 2015. doi:10.1109/TIME.2015.26.
9. C. Combi, M. Gambini, S. Migliorini, and R. Posenato, "Representing business processes through a temporal data-centric workflow modeling language: An application to the management of clinical pathways," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, pp. 1182–1203, Sept. 2014. doi:10.1109/TSMC.2014.2300055.
10. C. Combi, M. Gozzi, R. Posenato, and G. Pozzi, "Conceptual modeling of flexible temporal workflows," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 7, pp. 191– 1929, July 2012. doi:10.1145/2240166.2240169.

Verona, 2020-01-29