Linköping Electronic Articles in
Computer and Information Science
Vol. 2(1997): nr 2

Contingent durations in temporal CSPs: from consistency to controllabilities

Thierry Vidal
Department of Computer and Information Science
Linköping University
Linköping, Sweden

Hélène Fargier
Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier
Toulouse, France

Linköping University Electronic Press
Linköping, Sweden

http://www.ep.liu.se/ea/cis/1997/002/

Published on January 22, 1997 by
Linköping University Electronic Press
581 83 Linköping, Sweden

Linköping Electronic Articles in
Computer and Information Science
ISSN 1401-9841
Series editor: Erik Sandewall

Typeset by the authors using LaTeX
Formatted using étendu *style*

Recommended citation:
<Authors>. <Title>. Linköping Electronic Articles in Computer and Information Science, Vol. 2(1997): nr 2. http://www.ep.liu.se/ea/cis/1997/002/. *January 22, 1997.*

This URL will also contain a link to the authors' home pages.

Abstract

A Temporal CSP is a network of time-points related by binary constraints expressing maximal and minimal durations between them. This formalism has proven to be useful in many different research areas, but it has also proven to be limited in the sense that it does not take into account the *contingent* nature of some constraints: in many real applications, we cannot decide some effective durations that are not under our control but will be provided by the external world. This paper aims at proposing an extension of TCSP that enables the expression of such constraints. Here indeed the classical consistency property must be redefined in terms of *controllability* of the network : in outline, a network is controllable iff it is consistent in the classical sense in any *situation* that could arise in the external world, i.e. whatever the actual durations of the contingent intervals are. A more in-depth analysis leads to the identification of three different levels of controllability, the *Strongs*, the *Weak* and the *Dynamic* one. The paper will focus on the representation and concept issues, with some hints on their relevance in different domains. The reasoning aspects (complexity and algorithms) will only be sketched in this preliminary report.

This paper is under submission to TIME'97 workshop.

1 Background and overview

A large number of research domains in A.I. need to handle time in an explicit and highly expressive manner. Apart from studies dealing with logics of time, researches have been carried out on temporal algebras, i.e. formalisms solely capturing the time entities (points or intervals) and the relations among them. Constraint Satisfaction Problems model them in terms of variables and constraints. We should distinguish here between *symbolic* constraint algebras [1, 15], models dealing with *numerical* constraints[1] [4], and some advanced proposals combining both [12, 9, 6].

We will focus in this paper on numerical temporal constraints in the *Temporal CSP (TCSP)* formalism [4]. Those are useful in such domains as scheduling [7], supervision [5], diagnosis and temporal databases [3], multimedia authoring environments [10], or planning. In this latter domain, the incremental planner IxTet [9] used a TCSP model to check the temporal consistency of the plan (i.e. a partially ordered set of tasks allowing an agent to reach a given goal). The next step [14] was to take into account the inherent uncertain nature of durations of some tasks in realistic applications, distinguishing between *contingent* constraints (whose effective duration will only be observed at execution time. e.g. the duration of a task) and *free* ones (which instanciation is controlled by the agent, e.g. the duration between starting times of tasks). This will be recalled in Section 2.

Inspired by the thorough study carried out in the framework of discrete CSPs [8] on the same kind of distinction, we had to redefine consistency in terms of what we called the *Strong controllability* (i.e. existence of one solution that will satisfy the whole set of constraints in any situation possibly arising in the external world) and the *Weak controllability* (i.e. in any situation, there exists at least one solution that satisfies the constraints in that context). But these notions fail unfortunately to encompass the reactive nature of the solution building process in dynamic domains like planning, which Section 3 will explain and argue, eventually issuing the thorough definition (which is the main contribution of this article) of a third level of controllability, the *Dynamic* one. Section 4 will only quickly and partially tackle the reasoning issues (complexity, tractable subclasses and algorithms).

2 Representation issues

2.1 The Temporal CSP

In the time-point continuous algebra, time is represented through a set of time-points related by a number of relations (see [15]), that can be be represented through a graph of time-points and precedence

[1]The unified term of *duration* will be used for designing the value of such constraints, possibly expressing as well notions of dates and delays.

($\preceq$) relations [9]. One can also use time-point graphs to represent numerical constraints, thanks to the TCSP formalism [4]. Here binary constraints define the possible durations between two time-points by means of temporal intervals. The *STP* (Simple Temporal Problem) restriction of general TCSPs applies when those are only non-disjunctive intervals: $c_i = [l_i, u_i]$ between x and y expresses that the values of x and y must be such that $(y\text{-}x) \in [l_i, u_i]$. A TCSP is said to be consistent if one can *choose* for each time point a value such that all the constraints are satisfied, the resulting instanciation being one *solution* of the TCSP. Consistency checking of a general TCSP is NP-complete, but for some restrictions such as the STP, polynomial-time polynomial (e.g. the PC-2 3-consistency checking algorithm [11]) are complete.

2.2 The STPU

The TCSP model suits well the cases where effective durations are under the control of the agent. If not, the problem has to be redefined in the following way [14]:

Definition 2.1 (Two kinds of constraints / variables)
A free constraint c_i *(referred as* Free *in the following) is a numerical constraint of type* $(\text{x-y}) \in [l_i, u_i]$ *that will be instanciated* by the agent *in the domain* $[l_i, u_i]$.

A contingent constraint g_i *(referred as* Ctg *in the following) is a numerical constraint of type* $(e_i - b_i) \in [l_i, u_i]$ *that will be instanciated* by the external world *in the domain* $[l_i, u_i]$.

The activated *time-points* b_i *are those which date is assigned by the agent.*

The received *time-points* e_i *are those which unpredictable date is assigned by the external world.*

Hence the domain of a *Free* can be reduced by propagation (removing values that are inconsistent with other contraints [4]) while the domain of a *Ctg* should NOT (as it would remove possibly occurring values). Those distinctions allow the definition of a new model called *Simple Temporal Problem under Uncertainty (STPU).*

Definition 2.2 (STPU) $\mathcal{N} = (V_b, V_e, R_g, R_c)$ *represents a* STPU *with*

$V_b = \{b_1, \ldots, b_B\}$ *is the set of the* B *activated time-points,*
$V_e = \{e_1, \ldots, e_E\}$ *is the set of the* E *received time-points,*
$R_c = \{c_1, \ldots, c_C\}$ *is the set of the* C *Frees,*
$R_g = \{g_1, \ldots, g_G\}$ *is the set of the* G *Ctgs, with* $\forall g_i = [l_i, u_i], l_i > 0$.

So a Ctg encompasses the notion of an activity (e.g. a task) which (non-null) duration is contingent. One can notice that those definitions entail the basic property that a Ctg always relates an *activated* time-point ("begin") to a *received* one ("end"), and that hence two activated [resp. received] time-points will always be related by a Free.

3 Consistency revisited: 3 levels of controllability

3.1 Preliminary definitions

Definition 3.1 (Control sequence, Current-solution)
A control sequence δ *of the STPU is an assignment of the sole activated time-points:* $\delta = \{b_1, \ldots, b_B\}$[2]*.*
A current-solution *at time-point i is a partial control sequence* $\delta_{\prec i} = \{b_1, ..., b_{B'}\}$, *B' < B, such that* $\forall j = 1 \ldots B'$, $b_j \prec i$.

Hence, in planning for example, where the solution is built reactively, a *current-solution* defines that part of *a control sequence* (i.e. a starting time for each task) currently developped until point i. This entails that $\forall \delta$ and $\forall i = 1 \ldots B$, $\exists! \delta_{\prec i} \subseteq \delta$.

Definition 3.2 (Situation) *Given that* $\forall i = 1 \ldots G$, $g_i = [l_i, u_i]$, $\Omega = [l_1, u_1] \times \ldots \times [l_G, u_G]$ *will be called the* space of situations, *and* $\omega = \{\omega_1 \in [l_1, u_1], \ldots, \omega_G \in [l_G, u_G]\} \in \Omega$ *will be called a* situation *of the STPU.*

Definition 3.3 (Projection) $\forall \omega \in \Omega$, $\mathcal{N}_\omega$ *is called the* projection *of the STPU* $\mathcal{N}$ *in the situation* ω*, and is constructed by replacing every* Ctg $g_i = [l_i, u_i]$ *in* $\mathcal{N}$ *by the singleton* $g_i = \{\omega_i\}$, *with* $\omega_i \in \omega$.

Property 3.1 *A* projection $\mathcal{N}_\omega$ *defines a classical STP* [14].

Definition 3.4 (Past-situation, Situation-to-come) *In the same way as we have defined a current-solution, we can define the* past-situation $\omega_{\prec i} \in \Omega_{\prec i}$ *[resp. the* situation-to-come $\omega_{\succ i} \in \Omega_{\succ i}$*] at point i as being the set of observations*[3] *prior to [resp. subsequent to] i.*

3.2 The Strong controllability

Definition 3.5 (Strong controllability) $\mathcal{N}$ *is* Strongly controllable
iff
$\exists$ *a control sequence* $\delta = \{b_1, ..., b_B\}$ *such that* $\forall \omega \in \Omega, \delta$ *is a solution of* $\mathcal{N}_\omega$.

In other words, the STPU is *Strongly controllable* iff there exists at least one "universal" solution that fits any situation (and that hence, as far as dynamic applications are concerned, might be computed off-line beforehands).

[2] Notice that in order to ease the formulation, we adopt the usual misuse of merging variables and values into one unique notation.

[3] Intuitevily, an *observation* is the effective duration ω_i received by the system at time-point e_i, whereas a *decision* will refer to the value assigned to an activated time-point b_i.

Examples

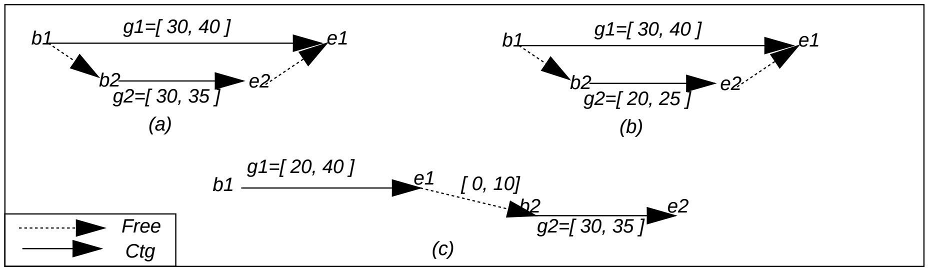

Figure 1: Strong controllability checking

The two first drawings in figure 1 show a *during*-like relation [1] between two Ctgs. The first case (a) is not Strongly controllable, since there exists at least one situation $\{\omega_1{=}30,\ \omega_2{=}35\}$ that violates the *during* relation. The second case (b) is Strongly controllable since for instance the control sequence $\{b_1 = 0, b_2 = 0\}$ is valid whatever values are taken by ω_1 and ω_2.

The third case (Figure 1(c)) exhibits a simple *before*-like relation [1] between two Ctgs that would be rejected by a Strong controllability checking algorithm, since instanciation of b_2 depends on ω_1. But it looks controllable from a planning point of view since one will have no problems deciding when to activate the second "task" g_2 once the first one g_1 is achieved. Hence, Strong controllability appears to be a too much demanding property, calling for a "weaker" one ....

Anyway, Strong controllabilty may be relevant in specific applications where the situation is not observeable at all or where the complete control sequence must be known beforehands (e.g. when other activities processed by other agents depend on it, which for instance may arise in the production planning area).

3.3 The Weak controllability

Definition 3.6 (Weak controllability) *$\mathcal{N}$ is* Weakly controllable *iff*
$\forall \omega \in \Omega,\ \exists \delta = \{b_1, ..., b_B\}$ *such that δ is a solution of $\mathcal{N}_\omega$.*

In other words, the STPU will remain *Weakly controllable* as far as in any given situation, there exists at least one solution (which holds in the case of figure 1(c)). Hence, as soon as one knows the situation, one can pick and apply the control sequence that matches the situation.

Example

Figure 2 exhibits a possible 3-Ctgs network, which is shown to be Weakly controllable (this will not be proven here). Unfortunately, in dynamic domains such as planning, the solution (the plan) is applied

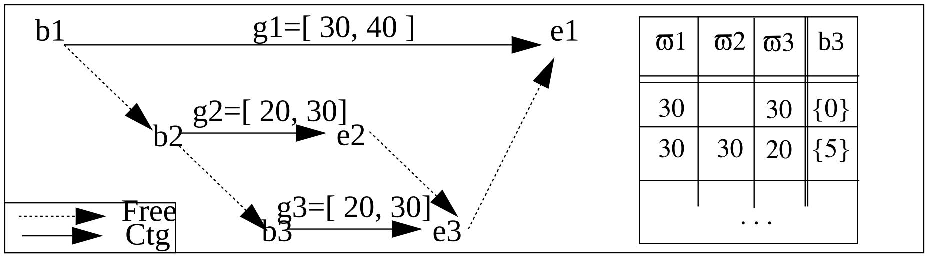

Figure 2: Shortcomings of the Weak controllability

(executed) reactively, i.e. whereas the situation remains partially unknown. For instance, if we assume that $b_1 = b_2 = 0$ (which will always ensure an optimal control sequence), then the "sequence" to find is reduced to the simple decision of instantiating b_3. The table next to the figure exhibits two distinct situations where one can see that the decision to be taken depends upon the observation of ω_3, which will remain unknown when b_3 is activated. Therefore in dynamic domains we would like to conclude that the network is "not controllable".

Hence the property that we seek for our planning application is not Weak controllability either. But again, this property may be relevant in specific applications where the situation will be known JUST BEFORE the execution starts (consider for instance delivering delays in production planning), but one wants to know in advance that there will always be at least one feasible solution.

3.4 The Dynamic controllability

In dynamic application domains, the decisions must be taken in the process of time, as far as one "observes" the situation at hand. This means that at each point in time, the situation splits into the *past-situation* that is known and the *situation-to-come* that remains unknown.

Definition 3.7 (Dynamic controllability) $\mathcal{N}$ *is* Dynamically controllable <u>*iff*</u>
$\forall \omega \in \Omega$, $\overline{\overline{\exists}}\delta = \{b_1, ..., b_B\}$ *such that*
$\forall i \in V_b$ *with past-situation* $\omega_{\prec i} \subset \omega$, $\forall$ *possible situation-to-come* $\omega'_{\succ i} \in \Omega_{\succ i}$, $\exists\ \delta'$ *such that the* current-solution $\delta_{\prec i} \subseteq \delta'$ *and* δ' *is a solution of* $\mathcal{N}_{\omega_{\prec i} \cup \omega'_{\succ i}}$.

In other words, each successive decision is ensured to extend to a global solution whatever occurrences remaining to be observed. Planning is not the only domain in which this applies: in multimedia authoring environments research, [10] make the same distinction betwen contingent and free constraints to model the temporal structure and dynamic presentation of a document. It appeared recently that our *Dynamic controllability* was the one to be considered in this framework.

3.5 The basic property relating the three levels

Property 3.2
Strong controllability ⇒ Dynamic controllability ⇒ Weak controllability.

Sketch of proof: The first implication is straightforward: if there exists a "universal" control sequence δ, then of course in any situation, we can choose to develop this one, and at any point in time, the current solution extends to δ itself in any possible situation-to-come.
The second implication is even more trivial: for any situation ω, if a current-solution can be extended to any situation-to-come, then it will be the case in the situation ω itself.

In other words, when trying to satisfy the Dynamic controllability requirement, checking the Strong one will be a complete but not sound process, whereas Weak controllability checking will issue sound but uncomplete answers.

4 Complexity issues and reasoning techniques

This section will not be deeply developped (and some proofs will not be provided) in this preliminary report, because of the limited length and also because it refers to already published or on-going work.

4.1 Checking the Strong controllability

For an in-depth analysis of the Strong controllability checking, we invite the reader to refer back to the *Decision Graph* method described in [14]. We will only recall hereafter the basic tractability property.

Property 4.1 (Complexity of the Strong controllability)
Checking the Strong controllability is polynomial.

Sketch of proof: The problem of deciding Strong controllability of a STPU can be represented by means of a classical STP such that the STPU is Strongly controllable iff the STP is consistent in the classical sense. The idea is to consider the relationships between tasks in the worst case, assuming that a contingent duration $d_i \in [l_i, u_i]$ is equal to u_i in any constraint of the form $x - y > d_i$ and equal to l_i in any constraint of the form $x - y < d_i$. Hence, since determining whether a STP is consistent or not is a polynomial problem, so it is for deciding Strong controllability.

4.2 Checking the Weak controllability

Conjecture 4.1 (Complexity of the Weak controllability)
Checking the Weak controllability is Co-NP-complete.

Sketch of proof: The Co-problem of checking Weak controllability is: is there a situation $\omega \in \Omega$ such that $\mathcal{N}_\omega$ is an inconsistent STP ? Since checking that a STP is inconsistent is a polynomial problem, this co-problem belongs to NP. Hence, Weak controllability belongs to Co-NP. The difficulty of the problem (Co-NP-*complete*) remains to be proven.

Property 4.2 *if* $\mathcal{N}_{\{\omega_1,\ldots,l_i,\ldots,\omega_G\}}$ *and* $\mathcal{N}_{\{\omega_1,\ldots,u_i,\ldots,\omega_G\}}$ *are consistent STPs, then, for any* $v_i \in [l_i, u_i]$, $\mathcal{N}_{\{\omega_1,\ldots,v_i,\ldots,\omega_G\}}$ *is a consistent STP.*

Property 4.3 *A STPU is Weakly controllable iff for any* $\omega^{bnd} \in \{l_1, u_1\} \times \ldots \times \{l_G, u_G\}$, $\mathcal{N}_{\omega^{bnd}}$ *is a consistent STP.*

If Weak controllabilty proves to be co-NP complete, one can now imagine an enumerative algorithm which checks the consistency of the projection $\mathcal{N}_{\omega^{bnd}}$ for every $\omega^{bnd} \in \{l_1, u_1\} \times \ldots \times \{l_G, u_G\}$. It can be processed recursively:

> Consider $\mathcal{N}$ as a classical STP, and propagate the constraints by 3-consistency: if a Ctg $g_i = [l_i, u_i]$ is reduced, it means that at least one projection $\mathcal{N}_{\omega^{bnd}}$ is inconsistent and therefore the STPU is not Weakly controllable. Otherwise, choose one g_i and work recursively on the two simplified STPUs obtained for $g_i = \{li\}$ and for $g_i = \{u_i\}$[4]. The resulting complexity of this (probably non-optimal) algorithm is in *O*(2^G).

4.3 Checking the Dynamic controllability

The problem of characterizing the complexity of the Dynamic controllability seems to be a non-trivial problem. We are mainly concerned with merely proving that it is not polynomial. This can be made by trying to prove that Dynamic checking is more complex than Weak controllability checking, which is probably non-polynomial. This proof is currently being carried out.

The next step will be to look for (a priori exponential) algorithms. The first idea is to design an uncomplete but efficient one only evaluating the feasibility of the whole "plan", which could be incrementally reevaluated during the execution, together with a complete (but exponential) controllability checking algorithm only applied in the short-range, in a reactive manner. The problem could hence be seen as a "game against the nature". The second idea is to take advantage of the structure of the temporal graph to design graph-search "simulation" algorithms that would fall back under the polynomial class in specific areas (like for instance in [10]).

4.4 Equivalence classes: a quick overview

Assuming that Weak and Dynamic controllabilities are more than polynomial, and that the conjecture NP $\neq$ P is true, then it is always useful to track maximal restricted representations in which the problem becomes again polynomial [13, 2, 6]. In our framework, it amounts to find the maximal equivalence classes between Weak [resp. Dynamic] and Strong controllabilities.

In [14], we have defined an algebra of the possible relations linking two Ctgs (for instance in Figure 1, the *during*-like relation will be

[4] Notice that in a look-ahead approach one can propagate the constraints after each instanciation.

written $g_2\mathcal{R}_8 g_1$ and the *before* one will be $g_1\mathcal{R}_4 g_2$) that was useful not only for building our Decision Graph, but also for proposing a first equivalence class between Weak and Strong controllabilities, restricting the possible set of relations $\mathcal{R}_i$. This first attempt suffered from uncorrect approximations, that called for a deeper analysis not yet fully achieved.

Our first extension concentrates on the Weak $\equiv$ Strong equivalence classes, keeping our definition of a precedence constraint between time-points as being $\preceq$ (before or equals) [15]. We have proven the following basic result, which allows to build maximal tractable subclasses in this context (not presented here).

Definition 4.1 (Upper / Lower Ctgs)
A Ctg $g_i = [l_i, u_i]$ *is said to be* lower <u>*iff*</u> $\forall\delta$ *solution of* $\mathcal{N}_{\{\omega_1,\ldots,l_i,\ldots,\omega_G\}}$, δ *is also a solution of* $\mathcal{N}_{\{\omega_1,\ldots,u_i,\ldots,\omega_G\}}$.

A Ctg $g_i = [l_i, u_i]$ *is said to be* upper <u>*iff*</u> $\forall\delta$ *solution of* $\mathcal{N}_{\{\omega_1,\ldots,u_i,\ldots,\omega_G\}}$, δ *is also a solution of* $\mathcal{N}_{\{\omega_1,\ldots,l_i,\ldots,\omega_G\}}$.

Property 4.4 ($\preceq$-Equivalence class Strong $\equiv$ Weak)
<u>*If*</u> *all the Ctgs* g_i *are* lower *[resp.* upper*],* <u>*then*</u> *Strong controllability is equivalent to Weak controllability.*

Another track of research consist in redefining the precedence relation between time-points as being now $\prec$ (strictly before) [15]. Not only this defines an algebra closer to Allen's one [1], but it gives back more expressive equivalence classes, both for the Weak and the Dynamic controllability (only the latter being drawn here):

Property 4.5 ($\prec$-Equivalence class Strong $\equiv$ Dynamic)
<u>*If*</u> $\prec$ *defines the precedence relation between time-points,*
<u>*Then*</u> $\exists$ *a maximal equivalence class between Strong and Dynamic controllability, obtained by imposing:*

any Free between a received time-point e_i *and an activated time-point* b_j *can only be such that* $b_j - e_i =]0, +\infty[$.

5 Conclusion

This preliminary report presented extensions of the STP model to take into account contingent durations (i.e. assignments provided by the external world). Of special interest is the new consistency property called *Dynamic controllability*, that suits well those application domains in which the solution is built in a reactive way. If the model and the concepts are now clear, it remains to prove the assumed hardness of the identified problems, and to propose algorithms to deal with it. Efficiency being a key issue in dynamic applications, a special interest is given in our present on-going work on the thorough characterization of maximal tractable classes and the design of efficient algorithms (uncomplete or possibly complete ones in specific applications, like in [10]) for checking the Dynamic controllability.

References

[1] J.F. Allen - *Maintaining knowledge about temporal intervals*, Communications of the ACM, 26(11):509–521, 1983.

[2] C.Bessière, A.Isli & G.Ligozat - *Global consistency in interval algebra networks: tractable subclasses*, In Proc. of the 12th European Conference on A.I. (ECAI-96), pp. 3–7, Budapest (Hungary), 1996.

[3] V.Brusoni, L.Console, B.Pernici & P.Terenziani - *LaTeR: a general purpose manager of temporal information*, Methodologies for Intelligent Systems 8. Lecture Notes in Computer Science 869:255–264, Springer Verlag, 1994.

[4] R.Dechter, I.Meiri & J.Pearl - *Temporal constraint networks*, Artificial Intelligence, 49:1–95, 1991.

[5] C.Dousson, P.Gaborit & M.Ghallab - *Situation recognition: representation and algorithms*, In Proc. of the 13th International Conference in A.I. (IJCAI-93), Chambéry (France), 1993.

[6] T.Drakengren & P.Jonsson - *Eight maximal subclasses of Allen's interval algebra with metric time*, to appear in Journal of Aritificial Intelligence Research (JAIR), 1996.

[7] D.Dubois, H.Fargier & H.Prade - *The use of fuzzy constraints in job-shop scheduling*, In Proc. of IJCAI-93 Workshop on Knowledge-Based Planning, Scheduling and Control, Chambéry (France), 1993.

[8] H.Fargier, J.Lang & T.Schiex - *Mixed constraint satisfaction: a framework for decision problems under incomplete knowledge* - In Proc. of the 12th National Conference on A.I. (AAAI-96), Portland (Oregon, USA), 1996.

[9] M.Ghallab & T.Vidal - Focusing on a sub-graph for managing efficiently numerical temporal constraints, in Proc. of FLorida A.I. Research Symposium (FLAIRS-95), Melbourne Beach (FL, USA), 1995.

[10] M.Jourdan, N.Layaïda & L.Sabry-Ismail - *Time representation and management in MADEUS: an authoring environment for multimedia documents*, to appear in Multimedia Computing and Networking, February 1997.

[11] A.K.Mackworth & E.C.Freuder - *The complexity of some polynomial network consistency algorithms for constraint satisfaction problems*, Artificial Intelligence, 25(1):65–74, 1985.

[12] I.Meiri - *Combining qualitative and quantitative constraints in temporal reasoning*, In Proc. of the 9th National Conference on A.I. (AAAI-91), Anaheim (CA, USA), 1991.

[13] B.Nebel & H.J.Bürckert - *Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra*, Journal of the ACM, 42(1):43–66, 1995.

[14] T.Vidal & M.Ghallab - *Dealing with Uncertain Durations in Temporal Constraint Networks dedicated to Planning*, In Proc. of the 12th European Conference on A.I. (ECAI-96), pp. 48-52, Budapest (Hungary), 1996.

[15] M.Vilain, H.A. Kautz & P.vanBeek - *Constraint Propagation Algorithms: a Revised Report*, Readings in Qualitative Reasoning about Physical Systems, Morgan Kaufman, 1989.