

Representation and reasoning with disjunctive temporal constraints

TRACK 1: Temporal representation and reasoning in AI

TOPIC: Temporal constraint reasoning

María Isabel Alfonso Galipienso

Depto. Ciencia de la Computación e Inteligencia Artificial

Universidad de Alicante(Spain)

e-mail:eli@dccia.ua.es

Federico Barber Sanchís

Depto. de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia(Spain)

e-mail:fbarber@dsic.upv.es

February 21, 2002

Abstract

The purpose of this paper is to show the expressiveness provided by the use of a Labelled TCSP model in order to specify and reason about disjunctive temporal constraints. We use a network based representation. The reasoning algorithms allows us to manage complex temporal constraints, both in assertion and query processes. It allows disjunctive assertions, conjunctive and hypothetical queries, and one-to-many constraints. Additionally, a labelled point-based metric model becomes an adequate support for reasoning on costs associated to the use of resources. They can be managed by an algorithm that integrates effectively a CSP (Constraint Satisfaction Problem) process into a closure process. The result is a new parameterizable process that can be applied to practical and real problems.

1 Introduction

A Temporal Constraint Satisfaction Problem (TCSP) is a particular class of CSP problem where variables represent times and constraints represent sets of allowed temporal relations between them. It requires a temporal model where the goals are to reason about what consequences(T) follow from a set of temporal constraints,

"Temporal-constraints $\models T$?", or to determine whether a set of temporal constraints is satisfiable, with no assumptions about properties of temporal facts. A temporal reasoning model is made up by a temporal algebra which determines the expressiveness of the temporal model, and by temporal reasoning algorithms.

Several temporal models have been defined in the literature, which can be considered under two main features [7]. The first one is the representation expressiveness: '*What temporal information can the model represent as constraints among temporal entities?*'. The second one is related to the temporal reasoning algorithms ('*how to reason on a given type of temporal constraints to obtain what of them can be true*') and their computational complexity. A clear trade-off between these features exists. The management of disjunctive constraints allows a higher level of expressiveness, but it implies a higher computational cost [14], [8].

The main classical disjunctive temporal models are [11] point-based, interval-based, and metric (quantitative) point-based models. Some efforts have been made to integrate qualitative and quantitative temporal information, but the full integration of qualitative and quantitative information requires managing qualitative and quantitative disjunctive constraints among time points, intervals and durations. Moreover, some application domains (scheduling, causal reasoning, etc.) need to manage disjunctive assertions of temporal constraints, conjunctive and hypothetical queries, one-to-many constraints, etc. This gives rise to the need to manage non-binary constraints.

In order to overcome the limitations outlined above, new models are investigated. In [5] it has been proposed a temporal model (TCSP), based on a labelled point-based disjunctive metric temporal algebra, which gives rise to a labelled-TCN (LTCN). This model is based on *constraint-associated labels* so that constraints can be related among them without using hyper-arcs.

In this paper, we intend to show the high expressiveness provided by the Labelled TCSP model, which allows us to specify and reason about a great variety of constraints, such as disjunctive assertions: conjunctive and hypothetical queries, other complex and non-binary time-point constraints, and cost of use of resources.

We also explain an additional reasoning algorithm that integrates a CSP process into a closure process, and introduces some parameters in the resulting process.

The rest of paper is organized as follows. In section 2, we introduce the labelled TCSP model used. In section 3, we show the expressiveness of labelled temporal algebra and present several types of complex and non-binary constraints that can be specified. In section 4, we talk about the reasoning algorithms used to manage effectively the constraints specified. In section 5, we refer to some applications and evaluation of the model. Finally, we remark the main conclusions of this work.

2 Temporal model of disjunctive constraints

In [5], it has been described a new-labelled temporal algebra, whose main elements are:

- *labelled disjunctive metric constraints* (l_{ij}). The general form of a constraint l_{ij} is: $(t_i \{([d_{ij,k}^-, d_{ij,k}^+], \{label_{ij,k}\}), \dots, ([d_{ij,n}^-, d_{ij,n}^+], \{label_{ij,n}\})\} t_j)^1$, with $d_{ij,k}^- \leq d_{ij,k}^+$, which means: $(t_j - t_i \leq [d_{ij,1}^+, d_{ij,1}^-]) \vee \dots \vee (t_j - t_i \leq [d_{ij,n}^-, d_{ij,n}^+])$. Each $[d_{ij,k}^-, d_{ij,k}^+]$ denotes an *elemental* or *canonical* constraint $ec_{ij,k}$.
- *label sets* associated to *canonical constraints* ($\{label_{ij,k}\}$). Each $ec_{ij,k}$, when it is asserted, has associated a label set $\{label_{ij,k}\}$ with $|\{label_{ij,k}\}| = 1$ that identifies it. We name the tuple $lec_{ij,k} = (ec_{ij,k}, \{label_{ij,k}\})$ as *labelled canonical* constraint. Each $label_{ij,k}$ can be considered as a unique symbol. It uses the special label 'R0' to denote that an input constraint l_{ij} has only one disjunct. Each $label_{ij,k} \in \{label_{ij,k}\}$ of a derived constraint provides information about which input ec_{ij} set has been asserted.
- sets of *inconsistent canonical constraints* (*I-L-Sets*). An (*I-L-Set*) is a set of labels $\{label_{ij,k}\}$ that represents a set of overall inconsistent canonical constraints. That is, they cannot all simultaneously hold. Canonical constraints $lec_{ij,k} \in l_{ij}$ are pairwise disjoint. Thus, each 2-length set of labels from each pair of $lec_{ij,k}$ is considered an *I-L-Set*, and is added to the super-set named *Inconsistence-Set* (*I-set*).

This model uses a LTCN (Labelled Temporal Constraint Network) based representation, and includes the special temporal points T0 and TF to indicate the beginning and ending of the world. The reasoning algorithms guarantee the consistency and obtain the minimal LTCN. These are the *updating process*, and a *total closure process* that infers new constraints from those explicitly asserted. They are based on the following operators:

- Temporal Inclusion (\subseteq_{lc}), that takes into account the inclusion of temporal intervals and the inclusion of their associated label sets.
- Temporal Union (\cup_{lc}), that performs the disjunctive temporal union of labelled constraints as the set-union of their canonical constraints. However, all labelled canonical constraints whose associated labels are *I-L-Sets* should be rejected.

¹we will use also the form $(t_i \{([d_{ij,1}^-, d_{ij,1}^+], \{label_{ij,1}\}), \dots, ([d_{ij,n}^-, d_{ij,n}^+], \{label_{ij,n}\})\} t_j)$

- Temporal Composition (\otimes_{lc}), and Temporal Intersection (\oplus_{lc}), which are based in the operations \otimes and \oplus defined in [9].

The associated label set of each derived canonical constraint references the set of explicitly asserted disjunctions from which the canonical constraint has been derived. Moreover, the reasoning process maintains inconsistent sets of explicitly asserted disjunctions in the set of *I-L-Sets*. It should be taken into account that labelled constraints, associated label sets and the set of *I-L-Sets* are jointly, integrally managed by the reasoning algorithms.

The computational cost of the *closure* process is $O(n^2 l^{2e})$, in which n is the number of nodes in the network, l is the maximal number of disjunctions of input constraints, and e is the number of input constraints updated in the previous LTCN. This is the bounded cost of each added problem constraint². This complexity makes infeasible to solve real problems, so we introduce a new parameterized reasoning algorithm that obtain one or several solutions (see section 4).

3 Complex and non-binary constraints

By reasoning on labelled disjunctive constraints, associated label lists and *I-L-Sets*, the TCSP model offers the capability of representing and managing non-binary disjunctive constraints. Particularly, logical relations among canonical constraints of different edges ($lc_{ij.x} \in lc_{ij}, lc_{kl.y} \in lc_{kl}$) can be specified. This feature will allow us to manage logical expressions of constraints between different pairs of nodes both in assertion and retrieval processes. Following, we will show how these logical expressions on metric constraints can be represented and managed in the proposed model. For reasons of simplicity, only two disjunctive elements are shown. However, more than two elements could be managed in a similar way:

- To represent that two canonical constraints ($lc_{ij.x} \in lc_{ij}, lc_{kl.y} \in lc_{kl}$) cannot hold simultaneously, that is $\neg(lc_{ij.x} \cup lc_{kl.y})$ the set of associated labels to $lc_{ij.x}$ and $lc_{kl.y}$, $\{R_{ij.x}, R_{kl.y}\}$, should be added to the set of *I-L-Sets*.
- To represent that two canonical constraints ($lc_{ij.x} \in lc_{ij}, lc_{kl.y} \in lc_{kl}$) should hold simultaneously, the Cartesian products $\{R_{ij.x}\} \times \{\{R_{kl.1}, \dots, R_{kl.q}\} - \{R_{kl.y}\}\}$ and $\{R_{kl.y}\} \times \{\{R_{ij.1}, R_{ij.2}, \dots, R_{ij.p}\} - \{R_{ij.x}\}\}$ should be added to the set of *I-L-Sets*.

²The best case occurs when the algorithm acts as a pure CSP. Then, the closure process has a polynomial cost.

- To represent a logical dependency between two canonical constraints, such as "If $lec_{ij.x}$ then $lec_{kl.y}$ " (where $lec_{ij.x} \in lc_{ij}, lec_{kl.y} \in lc_{kl}$), the Cartesian product $\{R_{ij.x}\} \times \{\{R_{kl.1}, R_{kl.2}, \dots, R_{kl.q}\} - \{R_{kl.y}\}\}$ should be added to the set of *I-L-Sets*.

3.1 Reasoning of sets of constraints and their consequences

Let us show a simple flow-shop scheduling example to illustrate some features about the retrieval of logical expressions from constraints and the consequences of such expressions. Three jobs $\{J_1, J_2, J_3\}$ share three resources $\{M_1, M_2, M_3\}$ in a given order (Table 1). The use of M_j by J_i gives rise to the operation O_{ij} .

Table 1: A scheduling example.

	Ready-time (Rt)	M_1	M_2	M_3	Due-time (Dt)
J_1	0 (rt_1)	10' (O_{11})	10' (O_{12})	10' (O_{13})	40' (dt_1)
J_2	0 (rt_2)	10' (O_{21})	10' (O_{22})	10' (O_{23})	50' (dt_2)
J_3	0 (rt_3)	10' (O_{31})	10' (O_{32})	10' (O_{33})	60' (dt_3)

Each operation O_{ij} can be represented by the temporal points O_{i1}^- and O_{i1}^+ , that denote the beginning and ending points of O_{ij} . We can specify this scheduling problem as the following sets of disjunctive temporal metric constraints:

- Ready-time constraints: $(T0 \{[rt_i, \infty[_{\{R_0\}}\} O_{i1}^-) \forall i=1..3$
- Flow-shop restrictions: $(O_{ij}^+ \{[0, \infty[_{\{R_0\}}\} O_{i(j+1)}^-) \forall i=1..3, j=1,2$
- Due-time constraints: $(T0 \{[0, (dt_j)]_{\{R_0\}}\} O_{i3}^-) \forall i=1..3$
- Disjunctive constraints: $(O_{ij}^+ \{[1, \infty[_{\{R_{ij,(i+k)j}\}}\} - \infty, 0]_{\{R_{j,i,j(i+k)}\}}\} O_{(i+k)j}^-)$
 $\forall i=1..3, \forall j=1,2,3, \forall k=1,2$, with $O_{ij}^+, O_{(i+k)j}^- \in O$ and $\{R_a, R_b\}$ is an *I-L-Set*. Each disjunctive label $R_{a,b}$ indicates a possible order between operations. For example the label $R_{12,22}$ reflects the fact that O_{12} is scheduled before O_{22} .

Given this constraint set, we apply the corresponding reasoning algorithms (see sections 2 and 5), and we obtain the minimal LTCN. It contains all feasible scheduling solutions and we are able to know all possible orders among operations, all starting or ending time of operations, etc. In addition, we obtain an *Inconsistence-Set (I-Set)* that contain sets of *I-L-sets*. This allows us to make queries about some

other restrictions that underlie the minimal LTCN, like non-binary constraints. For instance, we can determine whether a set of canonical constraints of different constraints are overall consistent by checking if the union of their label sets is not an *I-L-set*. As an example, suppose that the *I-set* associated to constraint sets of table 1 contains the *I-L-set* $\{R_0, R_{11,12}, R_{22,21}\}$. We can formulate the following questions:

- *It is possible for O_{11} to be before O_{12} and for O_{21} to be after O_{22} ?:* Under the common problem context (R_0), the answer is no, since the set of labels associate to these disjunctions $\{R_0, R_{11,12}, R_{22,21}\}$ is an *I-L-set*
- *It is possible for O_{11} to be before O_{13} , for O_{21} to be before O_{23} and for O_{21} to be after O_{22} ?:* The answer will be yes, if the set of labels associates to these disjunctions $\{R_{11,13}, R_{21,23}, R_{22,21}, R_0\}$ is not an *I-L-set* nor a superset of an existing *I-L-set*.

This kind of questions about the feasibility of a conjunctive sets of constraints between different pairs of time points can be solved without propagating their effects to all of the LTCN. Moreover, the consistency of the partial instantiation of a subset of LTCN variables in their domains can be assured if the union-set of the associated label sets to these instantiations is not an *I-L-set*. Thus, partial solutions can be assembled without having to propagate the partial instantiation to all LTCN. On the other hand, it is also important to know the implications of these choices. These implications can be considered as consequences of hypothetical queries on the LTCN: '*What happens if?*'. Thus, the consequences which follow from the fulfillment of a set of canonical constraints can be obtained by means of the associated label sets to these canonical constraints. For example: we ask if O_{22} can meet with O_{21} , and if O_{12} can meet with O_{11} due to some optimal criteria. Suppose that we have in the minimal LTCN: $(O_{22}^+ \{..., [0, 0]_{\{R_0, R_2, R_5, \dots\}}, \dots\} O_{21}^-)$, $(O_{12}^+ \{..., [0, 0]_{\{R_0, R_2, R_3, \dots\}}, \dots\} O_{11}^-)$. If the union set of labels $\{R_0, R_2, R_3, R_5, \dots\}$ is not an *I-L-set*, then we know that both of these choices can hold. In this case, we know also that these choices about the operations (O_{22}, O_{21}) and (O_{12}, O_{11}) imply that the associated disjunctions to the corresponding union-set of labels should also hold.

It must be taken into account that making certain choices about orders or assignment times for operations can prevent the feasibility of making other choices. Thus, we can easily know the feasibility of a certain decision, its consequences and which other choices (orders among operations, possible times, etc.) become unavailable by means of associated labels of such a decision and without having to update or propagate it. This feature is important in order to analyse and optimize feasible scheduling solutions.

3.2 Reasoning on complex time-point constraints

A specific kind of non-binary constraints (*disjunctive one-to-many constraints*) disjunctively restricts the temporal occurrence of a time-point with other time-points. For instance, a time point can be temporally restricted to the maximum/ minimum temporal occurrence of a set of time-points. These complex constraints are useful in scheduling problems where the ending time-point of each order should be associated to the ending time-point of the last task in the order. Moreover, these constraints can also be useful in 'reasoning about change' processes. In a typical causal relation ($C1, C2, C3 \rightarrow E$), the effect E holds while all causes $\{C1, C2, C3\}$ hold. Thus, the concluded effect E should be temporally constrained to the overlapping temporal interval where all causes $\{C1, C2, C3\}$ hold. That is, $E^- = \max(C1^-, C2^-, C3^-)$, $E^+ = \min(C1^+, C2^+, C3^+)$. Constraints of this kind, which also appear with non-disjunctive constraints, cannot be managed by usual models so that *ad hoc* procedures are needed [6], [13].

Thus, by using labelled temporal constraints, "the time-point t_f is at the maximum occurrence of the set of time points $\{t_1, t_2, \dots, t_n\}$ " can be represented as:

$$(\max(t_1, t_2, \dots, t_n)\{[0, 0]\}t_f) \equiv \forall t_i \in \{t_1, \dots, t_n\} : (t_i\{[0, 0]_{\{R_{Ai}\}}, [1, \infty]_{\{R_{Bi}\}}\}t_f)$$

and $\{R_{B1}, R_{B2}, \dots, R_{Bn}\}$ is an *I-L-Set*. The *I-L-Set* $\{R_{B1}, R_{B2}, \dots, R_{Bn}\}$ specifies that the time point t_f has the constraint $[0,0]$ with, at least, a time point t_i . Moreover, the constraint $(t_i\{[0, 0]\}t_f)$ can only exist between t_f and those time points t_i that also allow the constraint $[0,0]$ among them. Afterwards, when all time points become fully constrained on the time line, t_f becomes constrained with only the maximum time-points t_{max} : $(t_{max}\{[0,0]\}t_f)$. In another case, the constraint $(t_{max}\{[1, \infty]\}_{\{R_{Bi}\}}t_f)$ would fail.

The one-to-many constraint $(\max(t_1, t_2, \dots, t_n)\{[0,0]\}t_f)$ is bidirectional: (i) $T0$ becomes constrained at the maximum time-point of $\{t_1, t_2, \dots, t_n\}$, and (ii) by constraining the upper-bound of the occurrence of t_f , the upper-bounds of occurrences of the time-points $\{t_1, t_2, \dots, t_n\}$ also become constrained. For example, time-point t_f can be constrained at the end of the latest operation (use of the resource P3: $\{O_{13}, O_{23}, O_{33}\}$) of the scheduling problem of Table 1:

$$\begin{aligned} (\max(O_{13}^+, O_{23}^+, O_{33}^+)\{[00]\}t_f) &\equiv (O_{13}^+\{[00]_{R_{A1}}, [1\infty]_{R_{B1}}\}t_f), \\ &\quad (O_{32}^+\{[00]_{R_{A2}}, [1\infty]_{R_{B2}}\}t_f), \\ &\quad (O_{33}^+\{[00]_{R_{A3}}, [1\infty]_{R_{B3}}\}t_f), \end{aligned}$$

and $\{R_{B1}, R_{B2}, R_{B3}\}$ is an *I-L-Set*.

Without any more constraints in the problem, t_f becomes constrained ($T0\{[50, 60]\}t_f$), where $[50, 60]$ are the minimum and maximum possible due times for

scheduling. Thus, we can estimate the optimum scheduling due time as the lower bound of t_f . Moreover, we can restrict the constraint of t_f with T0 so that the resource/operation assignment times and the operation ordering also become more constrained. Therefore, we can force a desired scheduling due time in [50 60]. Otherwise, when all scheduling times become fixed, time point t_f will be also fixed to the final scheduling due time.

Similarly,

$$(min(t_1, t_2, \dots, t_n)\{[0, 0]\}t_f) \equiv \forall t_i \in \{t_1, \dots, t_n\} : (t_i\{[0, 0]\}_{R_{Ai}},]-\infty, -1]_{R_{Bi}}\}t_f)$$

and $\{R_{B1}, R_{B2}, \dots, R_{Bn}\}$ is an *I-L-Set*. Likewise, the constraint "no is at $[d_1^-, d_1^+]$ $[d_2^-, d_2^+]$, ..., $[d_n^-, d_n^+]$ from the maximum time point t_1, t_2, \dots, t_n " could also be represented:

$$(max(t_1, t_2, \dots, t_n)\{[d_1^-, d_1^+], [d_2^-, d_2^+], \dots, [d_n^-, d_n^+]\}t_f).$$

Finally, given the causal relation $I_1, I_2, \dots, I_n \rightarrow I_0$, the following constraints:

$$(max(I_1^-, I_2^-, \dots, I_n^-)\{[0, 0]\}I_0^-), (min(I_1^+, I_2^+, \dots, I_n^+)\{[0, 0]\}I_0^+),$$

restrict the interval I_0 to the temporal overlapping of intervals $\{I_1, I_2, \dots, I_n\}$. Moreover, the constraint $(I_0^-\{[1, \infty]\}I_0^+)$ implies that I_0 has a non-null duration, so that forces $\{I_1, I_2, \dots, I_n\}$ to overlap, which is an usual requirement in causal chaining processes.

3.3 Cost constraints

The labels of the model allows us to incorporate several additional information that can be managed in an integrated manner with the reasoning algorithms. This is important because we can specify more wide range of problems in the same manner and solve it without changing the reasoning algorithms applied. As an example of this feature, we show how specify cost associated to resources required to carry out the corresponding actions depending on the time in which the actions are carried out. The form of a cost constraint would be: "The resource r_k have a cost associated of x if it is used between the temporal points t_1 and t_2 ".

We denote the cost of use of the resource r_k by means the labelled temporal interval $[t_1, t_2]_{R_{ck}}$. In order to incorporate it into the network, we add the constraint $(t0\{[-\infty, t_1 - 1]_{R_a}, [t_1, t_2]_{R_{ck}}, [t_2 + 1, \infty]_{R_b}\}TF)$, in which R_a and R_b represent zero cost value [4]. Thus, heuristics applied can use label R_{ck} in order to calculate the corresponding operation costs. This will allow us to use the same resolution method to solve the CSP problem, with or without associated costs to actions.

4 Reasoning algorithms

In addition to reasoning algorithms outlined in section 2, a new reasoning process has been proposed in [2] that integrates effectively the CSP process into a limited closure process: not interleaving them but as a part of the same process. It is an iterative algorithm, in which we add a new constraint each time. It results in a flexible model that can be tailored by two parameters:

- The maximal number of indecisions maintained in the network.
- The variable and value heuristics used to prune the search space.

The first parameter allows us to perform as a pure CSP process (if the number of disjunctions maintained is equal to zero), as a pure closure process (if this number is not limited), or as a convenient mixed closure-CSP method. The greater the number of indecisions maintained, more solutions are also maintained, and fewer backtrackings are needed to obtain a solution. On the other hand, we obtain a more incremental method in that we reduce the need to know in advance all constraints. Moreover, the process varies itself automatically with the number of maintained disjunctions and which of them are maintained in every moment.

The second parameter, the set of heuristics to apply, provides efficiency to the method in that they can take better decisions. Each input constraint acts as a variable, and each disjunction in a constraint represents a possible value for that variable [12]. So, in each iteration we have to decide the next constraint to add (by means a variable heuristic), and which disjunctions (by means a value heuristic) have to be maintained in the network. We can use the named *mixed heuristics* [3], that combine the results of apply several heuristics to major information derived in the LTCN by the closure process. Moreover, when we add a disjunctive constraint, we can maintain several disjunctions, and leave (delay) the corresponding decisions for later, if we consider that the information supplied by the heuristics it is not sufficient to decide.

The computational cost of the *closure* process is $O(n^2 l^{2e})$, in which n is the number of nodes in the network, l is the maximal number of disjunctions of input constraints, and e is the number of input constraints updated in the previous LTCN. This is the bounded cost of each iteration of the algorithm *Closure-CSP*³. This complexity makes infeasible to solve real problems, so we introduce the parameter to maintain a maximal number of indecisions, and the use of convenient variable and value heuristics.

³The best case occurs when the algorithm acts as a pure CSP. Then, the closure process has a polynomial cost.

5 Applications and evaluation

Due to the reasoning algorithm outlined in section 4, we are able to apply it both in the framework of temporal reasoning (for example causal reasoning) and CSP problems [4]. For example, this model has been investigated under several types of scheduling problems, and it has been capable of specifying sets of constraints not contemplated in previous approximations, such as setup and maintenance periods of the resources, several work flows, consider the cost of use of resources, and other types of scheduling problems such as production lots [1].

On the other hand, the outlined mixed closure-csp process proposed results in an incremental reasoning process that is useful when temporal constraints are not initially known but are successively deduced from an independent process; for instance, in an integrated planning and scheduling system [10]. This process has been evaluated using both several instances from a known benchmark of scheduling problems, and randomly generated instances. The results obtained [2], [3], are promising in order to apply it to solve real problems.

6 Conclusions

We have shown the expressiveness of a Labelled TCSP model in order to specify and reason about disjunctive temporal constraints. This model allows us to:

- Represent and manage non-binary disjunctive constraints both in assertion and retrieval processes. More precisely, we have seen how to handle several types of *disjunctive one-to-many constraints*.
- Perform hypothetical queries and assemble partial solutions without having to propagate the partial instantiation to all LTCN. So, we can easily know the feasibility of a certain decision, its consequences and which other choices become unavailable by means of associated labels of such a decision. This feature is important in order to analyse and optimize feasible solutions.
- Incorporate information cost of use of resources, that can be managed in an integrated manner.

References

- [1] M.I. Alfonso. *Un modelo de integración de técnicas de clausura y CSP de restricciones temporales: aplicación a problemas de scheduling*. PhD thesis, Dep. Ciencia de la Comput. e I. A. Universidad de Alicante, Spain, 2001.

- [2] M.I. Alfonso and F. Barber. A mixed closure-csp method to solve scheduling problems. In *Proceedings of the 14th IEA/AIE 2001*, volume 2070 of *Lecture Notes in A. I.*, pages 559–570, Berlin, July 2001. Springer Verlag.
- [3] M.I. Alfonso and F. Barber. Nuevas heurísticas y medidas de textura para resolver problemas de scheduling mediante clausura y csp. In *Proceedings of the 9th CAEPIA'01*, volume 2, pages 833–842, Gijón, Spain, Nov 2001.
- [4] M.I. Alfonso and F. Barber. Complex constraints management. Submitted for publication in *ECAI'02*, Juny 2002.
- [5] F. Barber. Reasoning on interval and point-based disjunctive metric constraints in temporal contexts. *Journal of A. I. Research*, 122:35–86, 2000.
- [6] F. Barber, V. Botti, E. Onaindia, and A. Crespo. Temporal reasoning in reakt: An environment for real-time knowledge-based systems. *AI Communications*, 7(3):175–202, 1994.
- [7] L. Chittaro and A. Montanari. Trends in temporal representation and reasoning. *The knowledge engineering review*, 11(3):281–288, 1996.
- [8] R. Dechter. From local to global consistency. *Artificial Intelligence*, 55:87–107, 1992.
- [9] R. Dechter, I. Meiri, and Perl J. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.
- [10] A. Garrido, E. Marzal, L. Sebastiá, and Barber F. A model for planning and scheduling integration. In *Proceedings of the 8th CAEPIA'99*, pages 1–9, Murcia, Spain, 1999.
- [11] E. Schwalb and Ll. Vila. Temporal constraints: A survey. *Constraints*, 3(2-3):129–149, 1998.
- [12] K. Stergiou and M. Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence*, 120:81–117, 2000.
- [13] P. Terenziani. Integrating calendar dates and qualitative temporal constraints in the treatment of periodic events. *IEEE Trans. on knowledge an data engineering*, 9(5):763–783, 1997.
- [14] M. Vilain and H. Kautz. Constraint propagation algorithm for temporal reasoning. In *5th. National Conference on A.I.*, pages 377–382, Philadelphia, PA, 1986.