

# Reasoning with Disjunctive Fuzzy Temporal Constraint Networks

Alfonso Bosch<sup>1</sup>, Manuel Torres<sup>1</sup>, and Roque Marín<sup>2</sup>

<sup>1</sup>Dept. of Languages and Computation. University of Almeria.  
Campus Universitario. La Cañada (Almeria) 04120-E, SPAIN  
{abosch, mtorres}@ual.es

<sup>2</sup>Dept. of Information and Communications Engineering. University of Murcia.  
Campus de Espinardo. Espinardo (Murcia) 30100-E, SPAIN  
roque@dif.um.es

## Abstract

The Disjunctive Fuzzy Temporal Constraint Network (DFTCN) model is a framework that allows reasoning with fuzzy qualitative and quantitative complex temporal constraints. However, its general complexity is exponential. In this work, we review methods that can reduce this complexity, based on graph topology and preprocessing. The final goal is to integrate all these methods into a general tool for temporal reasoning with DFTCN.

## 1 Introduction

Fuzzy Temporal Constraint Network (FTCN) model, introduced in [1,12], allows to express simple temporal constraints, representing them by means of a convex and normalized possibility distribution. Fuzzy temporal constraints allow combining precise and imprecise information, which can be also qualitative and quantitative. Then, this model is suitable for temporal reasoning in domains where the combination of such constraint types is required. A fuzzy model allows intermediate consistency degrees, and to quantify the possibility and necessity of a relationship or query. And constraint propagation reduces one of the drawbacks associated with fuzzy reasoning, the degradation of distributions when chaining fuzzy rules.

In certain tasks, such as planning, a more general model is needed, where temporal constraints can be convex or not. Then, the FTCN model is enhanced, allowing determining a constraint with a finite set of possibility distributions, normalized and convex, obtaining the Disjunctive Fuzzy Temporal Constraint Networks (DFTCN) model. This model extends the TSCP framework proposed by Dechter [5], and it allows constraints such as “Irrigation is *much before* or *a little after* than Treatment”, and subsumes the Vilain & Kautz point algebra (PA) [20]. This framework allows to represent all the possible relationships between time points, between intervals and between time points and intervals, and their disjunctions.

The main drawback of DFTCN is its computational inefficiency, because generally these networks are non-decomposable networks [5], needing backtracking to find a solution. Determining the consistency and computing the minimal network are also exponential. With small problems, this is not a drawback, but in order to generalize the use of the model in a general scope, it would be interesting to simplify its processing, if possible. The idea is to explore different approaches to be used before applying backtracking.

One approach is to try avoiding backtracking, using the topology of the problem graph [6]. Another one is decomposing the network into subproblems that can be solved separately. And another one is to apply preprocessing, reducing the original network and testing the problem consistency.

This model is being integrated with an advisor system for the control of plagues in greenhouses<sup>1</sup>. The advisor system (SAEPI) makes a decision about the need of a treatment. The next step is the selection of the best treatment plan, based on a multicriteria decision model. This treatment consists usually of multiple actions, and these actions are affected by temporal constraints among them and with general cultivation scheduling. The last step is to use the DFTCN model to assess the suitability of the treatment plan, checking its consistency, and generating a schedule.

Next section introduces the DFTCN model; sections 3 and 4 present the different approaches; and section 5 includes the conclusions and future works.

## 2 Disjunctive Fuzzy Temporal Constraint Networks

**Definition 1.** A disjunctive fuzzy temporal constraint network (DFCTN)  $L^d$  consists of a finite set of  $n+1$  temporal variables  $X_0, \dots, X_n$  ( $X_0$  as time origin), whose domain is a full ordered set of equidistant precise instants  $\tau = \{t_0, t_1, \dots, t_i, \dots\}$  ( $t_0$  as time origin), and a finite set of disjunctive binary constraints  $L_{ij}^d$  among these variables. The separation between two consecutive instants,  $t_{i+1}-t_i$ , will be the maximum precision factor  $f_p$ , that can be selected depending on the nature of the problem.

**Definition 2.** A disjunctive binary constraint  $L_{ij}^d$  among temporal variables  $X_i, X_j$  is defined with a finite set of possibility distributions,  $\{\pi_{ij}^1, \pi_{ij}^2, \dots, \pi_{ij}^k\}$ , normalized and convex [7], defined over a set  $I$ , isomorphic with integers  $Z$ ; for  $n \in I$ ,  $\pi_m(n) \in [0,1]$  represents the possibility that a time quantity  $m$  can be precisely  $n$  time units.

A value assignation for variables  $X_i, X_j$ ,  $X_i=a; X_j=b$ ,  $a, b \in \tau$ , satisfies the constraint  $L_{ij}^d$  iff it satisfies one of its individual constraints:

$$\exists \pi_{ij}^p \in L_{ij}^d / \pi_{ij}^p(b-a) > 0 \quad (1)$$

---

<sup>1</sup> This work is partially supported by an EC FEDER Program grant (1FD97-0255-C03-03)

The maximum possibility degree of satisfaction of a constraint  $L_{ij}^d$  for an assignment  $X_i = a, X_j = b$  is

$$\sigma_{ij}^{\max}(a, b) = \max_{1 \leq p \leq k} \pi_{ij}^p(b - a) \quad (2)$$

A constraint  $L_{ij}^d$  among variables  $X_i, X_j$  defines a symmetric constraint  $L_{ji}^d$  among  $X_j, X_i$ . The constraints  $L_{ii}^d$  must be equal to the crisp element zero,  $\pi_0$ , defined as  $\pi_0(0) = 1$ , and  $\pi_0(m) = 0 \forall m \neq 0$ .

Unary constraints are represented using binary constraints among  $X_0$  and the constrained variable, and the network can be represented with a directed graph.

Disjunctive constraints  $L_{ij}^d$  must be expressed in a canonical form, disallowing the complete inclusion among the possibility distributions:

$$\forall p, q \exists m, o \in I / \pi_{ij}^p(m) > \pi_{ij}^q(m), \pi_{ij}^p(o) < \pi_{ij}^q(o) \quad (3)$$

The tightest constraint is the empty constraint  $\pi_\emptyset$ , given as  $\forall m \in I : \pi_\emptyset(m) = 0$ . If a DFTCN contains an empty constraint, it has no solution. The loosest constraint is the universal constraint  $\pi_U$ , given as  $\forall m \in I : \pi_U(m) = 1$ . The lack of a constraint is equivalent to the universal constraint.

**Example 1.** Consider an example of integrated pest management: a watermelon cultivation into a greenhouse. One month after planting, there is a plant louse plague that needs to be controlled. The recommended treatment is to release *Chrysoperla* (a predator) and to apply Bifentrin and Endosulfan (chemical products). However, Endosulfan has to be applied one week before of two weeks after Bifentrin, (because both products present a cross reaction) and *Chrysoperla* needs to be released not before four days after applying Bifentrin (because Bifentrin is toxic for *Chrysoperla*). In addition, the application of Endosulfan has to be done not before 3 days after release, or approximately one week after release.

Harvesting cannot begin until 15 days after the application of Endosulfan (due to toxicity for consumers), and harvesting season is approximately between two and two months and a half after planting.

In this problem, we can identify six events or temporal variables:

$X_0$ : Planting date	$X_1$ : Plant louse detection
$X_2$ : <i>Chrysoperla</i> releasing	$X_3$ : Bifentrin application
$X_4$ : Endosulfan application	$X_5$ : Harvest initial date

The set of defined temporal constraints is the following, taking a day as temporal unit, and using trapezoidal possibility distributions:

$$L_{01} = (30, 30, 30, 30) = \text{"A month after"}$$

$$L_{12} = (1, 1, \infty, \infty) = \text{"After"}$$

$$\begin{aligned}
L_{13} &= (1, 1, \infty, \infty) = \text{"After"} \\
L_{14} &= (1, 1, \infty, \infty) = \text{"After"} \\
L_{05} &= (56, 58, 77, 79) = \text{"Approximately between 2 and 2.5 months after"} \\
L_{32} &= (4, 4, \infty, \infty) = \text{"More than 3 days after"} \\
L_{34} &= \{(-7, -7, -7, -7), (14, 14, 14, 14)\} = \text{"One week before or two weeks after"} \\
L_{45} &= (15, 15, \infty, \infty) = \text{"15 or more days after"} \\
L_{24} &= \{(-11, -9, -5, -3), (4, 4, \infty, \infty)\} = \text{"Approximately one week before or more than 3 days after"}
\end{aligned}$$

Combining this information, we will be able to see if it is possible to use this combination of chemical products and natural predators (*consistency* of the problem), what are the allowed sequences and intervals of application (*solutions* of the problem), and to answer queries about temporal relationships between two actions, two intervals or one action and one interval. For constraint propagation, we need to define the intersection and composition operations.

**Definition 3.** Given  $L_{ij}^d, H_{ij}^d$ , defined among the same variables,  $X_i, X_j$ , described with the finite sets of possibility distributions  $\{\pi_{ij}^{L^k}, 1 \leq k \leq p\}$  and  $\{\pi_{ij}^{H^k}, 1 \leq k \leq o\}$ , we can define the intersection of the constraints  $L_{ij}^d$  and  $H_{ij}^d$  as the constraint  $Q_{ij}^d$ :

$$\pi_{ij}^{Q^m}(x) = \min(\pi_{ij}^{L^y}(x), \pi_{ij}^{H^z}(x)), 1 \leq m \leq p + o, 1 \leq y \leq p, 1 \leq z \leq o \quad (4)$$

**Definition 4.** Given a disjunctive binary constraint  $L_{ik}^d$  among the variables  $X_i, X_k$ , and other constraint  $H_{kj}^d$  among the variables  $X_k, X_j$ , we define the composition of  $L_{ik}^d$  with  $H_{kj}^d$  as the constraint  $Q_{ij}^d$ :

$$\pi_{ij}^{Q^x}(m) = \sup_{m=t+s} \min(\pi_{ik}^{L^y}(t), \pi_{kj}^{H^z}(s)), 1 \leq y \leq p, 1 \leq z \leq o, 1 \leq x \leq p * o \quad (5)$$

**Definition 5.** The set  $S$  of possible solutions of a DFTCN  $L^d$  is defined as the fuzzy subset from  $\tau^n$  associated to the possibility distribution given as:

$$\pi_S(t_1, \dots, t_n) = \min_{\substack{0 \leq i \leq n \\ 0 \leq j \leq n}} (\sigma_{ij}^{\max}(t_i, t_j)) \quad (6)$$

The possibility of that a  $n$ -tuple may be a solution is the lowest possibility degree obtained when satisfying each constraint.

**Definition 6.** A  $n$ -tuple  $T = (t_1, \dots, t_n) \in \tau^n$  of precise instants is an  $\sigma$ -possible solution of a DFTCN  $L^d$  if  $\pi_S(T) = \sigma$ .

**Definition 7:** A DFTCN  $L^d$  is  $\sigma$ -consistent iff its set  $S$  of possible solutions verifies that  $\sup_{T \in \tau^n} \pi_S(T) = \sigma$ .

We say that a DFTCN  $L^d$  is consistent if it is 1-consistent; and it is inconsistent if it does not have any solution ( $\sigma = 0$ ).

**Example 2.** For the treatment problem described in Example 1,  $T=(0,30,35,31,45,60)$  is a 1-possible solution. The sequence of treatment actions is: first, to apply Bifentrin (next day after pest detection, if possible); next, to release Chrysoperla (four days after Bifentrin); and finally, to apply Endosulfan (two weeks after Bifentrin). Harvesting can begin exactly two months after planting.

Constraint propagation reduces the domain of temporal variables, giving an equivalent network suppressing values that do not lead to any solution, and it can detect if the problem is inconsistent. But we need the most depurated representation of the original problem, and this is the *minimal network* [2], where all domain values for each variable (defined with the constraint between  $X_0$ ) belong to a possible solution. We need this minimal network to answer queries about the relationship between temporal entities.

Determining the consistency of a DFTCN is NP-hard, and the minimal network cannot be obtained in polynomial time. In addition, a DFTCN is not decomposable. Then, a backtracking search algorithm must be applied to obtain a solution. These results are proven in [5].

A DFTCN can be treated as a set of simple FTCN, generated selecting for each constraint one of its possibility distributions (a *labelling*) [5]. Simple FTCN are polynomial [14], and we could obtain the minimal network of a DFTCN generating and solving all the underlying FTCN [2].

If  $p$  is the maximum of possibility distributions in each constraint, and the network has  $q$  disjunctive constraints and  $n$  variables, then the minimal network  $M^d$  of a DFTCN  $L^d$  can be obtained with a complexity  $O(p^q n^3)$ . This approach can be improved with chronological backtracking [5], and it allows using complementary techniques to reduce the work of backtracking algorithms [10,21]. The study of the application of these techniques to DFTCN is not addressed in this work.

Once computed the minimal network, we can solve temporal queries about the relationship between two temporal variables,  $X_i$  and  $X_j$ , that represent the underlying process for computing relationships between temporal entities, as defined in [13] for FTCN model. The relationship represents a temporal distance between the two variables. In DFTCN, a relationship  $R^d$  can include disjunctions, and it must be expressed with a set of possibility distributions:  $\{\pi_R^1, \pi_R^2, \dots, \pi_R^p\}$ , representing each one a temporal distance.

We need to obtain the compatibility between the minimal constraint  $M_{ij}^d = \{\pi_{Mij}^1, \pi_{Mij}^2, \dots, \pi_{Mij}^o\}$  and the proposed relationship  $R^d$ . This compatibility can be estimated with the *possibility* and *necessity* degrees of  $R^d$  with  $M_{ij}^d$ , and they are defined as

$$\Pi(X_i R^d X_j) = \min(\max(\max_{1 \leq y \leq p} \min(\max_{1 \leq z \leq o} \min_{m \in I} (\pi_R^y(m), \pi_{Mij}^z(m)))))) \quad (7)$$

$$N(X_i R^d X_j) = \min(\max(\min_{1 \leq y \leq p} \max_{1 \leq z \leq o} \min_{m \in I} (\pi_R^y(m), 1 - \pi_{Mij}^z(m)))))) \quad (8)$$

The external min operator scans all the disjunctions of the relationship  $R^d$ , and the external max operator scans all the disjunctions of the minimal constraint  $M_{ij}^d$  for each disjunction of the relationship.

**Example 3.** Table 1 shows the minimal network for the Example 1. All the original constraints have been modified, except  $L_{01}$ , because it is a fixed fact (the plague detection). For instance, the two possibilities for  $L_{34}$  have been reduced to one (two week after).

**Table 1.** Minimal network for the DFTCN of Example 1, obtained with backtracking

	$X_0$	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$
$X_0$	(0,0,0,0)	(30,30,30,30)	(35,35,71,75)	(31,31,48,50)	(45,45,62,64)	(60,60,77,79)
$X_1$	(-30,-30,-30,-30)	(0,0,0,0)	(5,5,41,45)	(1,1,18,20)	(15,15,32,34)	(30,30,47,49)
$X_2$	(-75,-71,-35,-35)	(-45,-41,-5,-5)	(0,0,0,0)	(-25,-23,-19,-17) (-10,-10,-4,-4)	(-11,-9,-5,-3) (4,4,10,10)	(4,6,42,44)
$X_3$	(-50,-48,-31,-31)	(-20,-18,-1,-1)	(4,4,10,10) (17,19,23,25)	(0,0,0,0)	(14,14,14,14)	(29,29,46,48)
$X_4$	(-64,-62,-45,-45)	(-34,-32,-15,-15)	(-10,-10,-4,-4) (3,5,9,11)	(-14,-14,-14,-14)	(0,0,0,0)	(15,15,32,34)
$X_5$	(-79,-77,-60,-60)	(-49,-47,-30,-30)	(-44,-42,-6,-4)	(-48,-46,-29,-29)	(-34,-32,-15,-15)	(0,0,0,0)

Now, we can query if Chrysoperla release ( $X_2$ ) can be *two or more days after* or *before* ( $R^d$ ) Endosulfan application ( $X_4$ ). If “two or more days after”=(2,2,  $\infty$ ,  $\infty$ ), and “before”= $(-\infty,-\infty,1,1)$ , the the possibility and necessity degrees are 1, showing that the relationship is completely possible, and that there is not another opposite situation.

### 3 Graph Theory Based Approaches for DFTCN

Topological characteristics of constraint networks can help us to select more effective methods to solve them, and there are previous studies about this topic [4,6]. These characteristics have been examined for both FTCN and DFTCN models; in this work, we will focus only in topics involved with disjunctive problems. These are series-parallel networks, adaptive consistency, cycle-cutset method, decomposition into nonseparable components and tree-decomposition.

If a DFTCN is series-parallel [23], the path consistent network is the minimal network, although the intersection and composition operations are non-distributive. As a subproduct of checking if a network is series-parallel, a variable ordering is obtained, when deleting the nodes. Applying directional path-consistency (DPC) algorithm [6] in the reverse order, a backtrack-free network is obtained, and the minimal constraint between the first two variables

of the ordering too. This can be interesting when we need only to compute a minimal constraint for two variables, and not the minimal network, as in LATER [3]. In addition, if the network is series-parallel, we can decide absolutely if the network is consistent, by applying DPC algorithm in the reverse order.

Adaptive consistence [5,6] can transform any network into a backtrack-free network for searching a solution, but it is difficult to estimate the complexity of this method [22], because it imposes k-consistency conditions [9], and to assess its utility for computing a backtrack-free solution. It is not oriented to computing the minimal network, but it can detect inconsistency (no solution is found).

The *cycle-cutset* method [4] tries first to solve the part of the network that, once instantiated, reduces the problem to a network with a tree structure. The method is oriented to searching solutions, and for DFTCN it is not suitable.

The decomposition into nonseparable components is a classical method for graph management [8]. It isolates independent cycles, and gives a tree superstructure of the graph. The application of this method to constraint graphs had been proposed by Dechter et al [5,6]. We can see that it is also suitable for DFTCN.

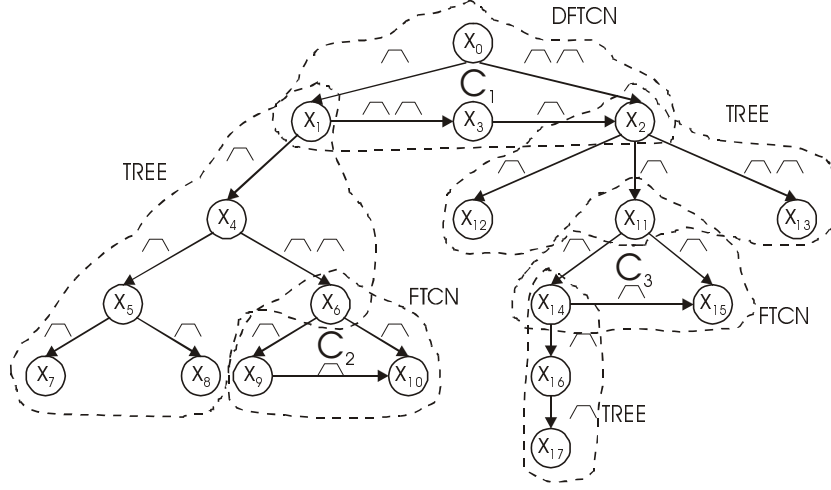
**Definition 8.** Let  $G=(V,E)$  the constraint graph of a DFTCN  $L^d$ , and let  $C=(V',E')$  be a nonseparable component from  $G$ . The minimal network of component  $C$ ,  $M_c^d$ , is the minimal network of the DFCTN defined by  $C$ .

**Theorem 2.** Let  $M^d$  the minimal network of a consistent DFTCN  $L^d$ , and let  $M_c^d$  the minimal network of a nonseparable component,  $C=(V',E')$ , from the constraint graph,  $G=(V,E)$ , of  $L^d$ . It accomplishes that  $\forall i, j \in V', x \in I, \max_{1 \leq k \leq p} \pi_{ij}^{M^k}(x) = \max_{1 \leq l \leq o} \pi_{ij}^{M_c^l}(x)$

**Proof outline:** The  $\leq$  direction is straightforward. For  $\max_{1 \leq k \leq p} \pi_{ij}^{M^k}(x) \geq \max_{1 \leq l \leq o} \pi_{ij}^{M_c^l}(x), \forall x \in I$ , we can show that for all  $x$  with  $\max_{1 \leq l \leq o} \pi_{ij}^{M_c^l}(x) > 0$ , there is a possibility value greater or equal in  $\max_{1 \leq k \leq p} \pi_{ij}^{M^k}(x) > 0$ .

From this result, we can try to reduce the overall complexity of the backtracking algorithm. First, we locate the nonseparable components,  $C_1, \dots, C_m$ , with a cost  $O(e)$ , and after, we solve each one of them individually. If all components are consistent, then the overall network is consistent, and the minimal networks of the components match with the complete minimal network. If we use backtracking, the maximum complexity is  $O(nr^3p^c)$ , where  $n$  is the number of nonseparable components,  $r$  is the highest number of nodes into components,  $c$  the highest number of arcs into components, and  $p$  the highest number of possibility distributions of a constraint.

The independency of minimal networks allows solving each component with the most efficient algorithm for each case. The components can be classified onto three types, as shown in Figure 1.



**Fig. 1.** Decomposition of a DFTCN into subnetworks. There are three kind of subnetworks: a) tree-structured (TREE), where each arc is a nonseparable component by itself; their constraints are minimal by definition. b) nonseparable components with cycles and without disjunctive constraints (FTCN); they are solved with an  $O(n^3)$  algorithm. c) nonseparable components with cycles and disjunctive constraints (DFTCN); they need to be solved with the backtracking algorithm.

Since the overall minimal network is not computed, additional computations are needed to obtain the minimal constraint between pairs of variables that belong to different components, and next theorem shows how to carry out this task.

**Theorem 3:** Let  $G=(V,E)$  be the constraint graph of a DFTCN  $L^d$ . Let  $i,j$  be two nodes belonging to different nonseparable components from  $G$ ,  $i \in C_i$   $y$   $j \in C_j$ . Let  $P$  be the only path  $P : C_i = C_{i_1}, a_1, C_{i_2}, a_2, \dots, a_k, C_{i_{k+1}} = C_j$  that connects  $C_i$  and  $C_j$  in  $G$  superstructure. Then  $M_{ij}^d = M_{i,a_1}^d \oplus M_{a_1,a_2}^d \oplus \dots \oplus M_{a_{k-1},a_k}^d \oplus M_{a_k,j}^d$

**Proof outline:** We need to see that  $M_{i,a_1}^d \oplus M_{a_1,a_2}^d \oplus \dots \oplus M_{a_{k-1},a_k}^d \oplus M_{a_k,j}^d \subseteq M_{ij}^d$  (reverse condition is true by definition of  $M_{ij}^d$ ).

The tree decomposition method consists of removing redundant constraints from the original network, until a tree that exactly represents the network without information loss is found [2,15,17]. This algorithm has a polynomial cost, and when it is applied to a minimal disjunctive network, it determines whether the network is decomposable or not. This schema has been implemented, and it is under evaluation with random and real problems [2].



## 4 Preprocessing techniques

When backtracking is unavoidable to solve a DFTCN, it is convenient to start from a network equivalent but more reduced than the original one. Removing useless information is called *preprocessing* a DFTCN, and there are several methods. The first group of them are path consistency algorithms, including PC-1 [16] and PC-2 [11], DPC [6] and the algorithm used for FTCN [14], that generalizes the Floyd-Warshall algorithm [5]. All these algorithms can be applied to DFTCN using the composition and intersection operations defined in section 2. Path consistency is a property of constraint satisfaction problems, which assures that the direct constraint between a pair of variables  $X_i, X_j$  does not allow a value forbidden by an induced constraint along a path  $P=\{X_i, X_k, X_j\}$ ,  $k \neq i, j$ . The second group includes algorithms for disjunctive constraints, as ULT and LPC [18,19], and they use modified operations that reduce the complexity, and avoid the fragmentation problem. These algorithms can also be extended to DFTCN. There are some variants of these algorithms [19], with lower levels of complexity and effectiveness.

All these algorithms compute a network equivalent to the original one, and, in addition, they can detect the inconsistency of a DFTCN, but not in all cases.

Preprocessing algorithms give as a result a complete network, and this is a drawback because it increases the backtracking effort [18]. When using the preprocessing result as input to backtracking, we propose to use only the constraints that map with the original problem, ignoring the induced constraints, because they only contain redundant information, as shown in this lemma.

**Lemma 1.** Let  $L^d$  be a DFTCN and  $G=(V,E)$  its constraint graph, and let  $H^d$  be the DFTCN obtained when preprocessing  $L^d$  with PC-1, PC-2, SPC, DPC, ULT, and LPC, represented with its graph  $G'=(V,E')$ . The DFTCN  $T^d$ , obtained by removing in  $G'$  the arcs  $\{E'-E\}$  is equivalent to  $H^d$  and  $L^d$ .

**Proof outline.** A preprocessed network  $H^d$  is equivalent to the original network  $L^d$ . The intersection of two equivalent networks results into another equivalent network. The intersection of  $L^d$  and  $T^d$  is equal to  $T^d$ , and  $T^d$  is included into  $L^d$ . The intersection of  $H^d$  and  $T^d$  is equal to  $H^d$ , and  $H^d$  is included into  $T^d$ . Then, as  $H^d$  is the most reduced of the three networks, it is equivalent to  $L^d$ , all three networks are equivalent.

This allows respecting the original topology when preprocessing, and it opens the possibility of combining preprocessing with graph-based techniques, in any order.

## 5 Conclusions and Future Work

In this work, we have presented the DFTCN framework for temporal reasoning with disjunctive fuzzy temporal constraints. Due to its exponential complexity, it is necessary to apply complementary techniques, and we have explored the possible application of different groups of techniques, defined for other constraint management frameworks: graph-based techniques, general path consistency techniques and special techniques for disjunction management, selecting the techniques summarized in Table 2.

**Table 2.** Complexity and features of candidate algorithms for managing DFTCN. Parameters: are  $n$ : problem variables;  $e$ : problem constraints;  $k$ : maximum number of possibility distributions in a constraint;  $R$ : problem range;  $q$ : number of disjunctive constraints;  $W^*(d)$ : maximum number of parents for a node on the induced graph

Algorithm	Complexity	Results and Observations
Chronological backtracking	$O(n^3 k^q)$	Minimal network & Solution Complete inconsistency detection
PC-2	$O(n^3 R^3)$	Path consistent network Partial inconsistency detection
DPC	$O(n W^*(d)^2 R^2)$	Directional path consistent network Partial inconsistency detection
SPC	$O(n^3 R^2)$	Approx. path consistent network Partial inconsistency detection
ULT	$O(n^3 ek + e^2 k^2)$	Approx. path consistent network Partial inconsistency detection & no fragmentation
LPC	$O(n^3 k^3 e)$	Approx. path consistent network Partial inconsistency detection & no fragmentation
SP	$O(n)$	Detect if a network is series-parallel Can lead to find backtrack-free solutions
CNS	$O(e)$	Decompose into nonseparable components
TD	$O(n^3 R^2)$	A tree equivalent to original network Can lead to find backtrack-free solutions Require a minimal or path consistent network

We have proposed a new use of the algorithm that solves FTCN networks for preprocessing DFTCN networks (SPC), and a decomposition into subnetworks that can be solved individually with the most effective algorithm for each case. We also showed that preprocessing does not increase the number of problem constraints, maintaining also the original graph topology, allowing technique combinations.

As future work, an immediate task is to implement all these algorithms for DFTCN, and evaluate its behaviour with real and random data, including them into a library. This library, with a selection mechanism, will be the core of a module for the representation and management of temporal information.

The selection mechanism could use initially heuristics based on direct problem features (topology, disjunction ratio, number of disjunctions, range).

In a next stage, we can study if it is possible to propose a cost model to bound “tractable” problems, and the necessary effort to solve a concrete problem.

## References

1. Barro, S., Marín, R., Patón, A.R.: A model and a language for the fuzzy representation and handling of time. *Fuzzy Sets and Systems*, 61 (1994) 153-175
2. Bosch, A., Torres, M., Navarrete, I., Marín, R.: Tree Decomposition of Disjunctive Fuzzy Temporal Constraint Networks. *FLA2001*, June 19-22, Bangor, Wales (2001), #1714-066, 7 pages
3. Brusoni, V., Console, L., B. Pernici, Terenziani, P.: LaTeR: a general purpose manager of temporal information, in: *Methodologies for intelligent systems 8*, LNCS 869 (1994) 255-264
4. Dechter, R.: Enhancement Schemes for Constraint Processing: Backjumping, Learning and Cutset Decomposition. *Artificial Intelligence* 41 (1990) 273-312
5. Dechter, R., Meiri, I., Pearl, J.: Temporal constraint networks. *Artificial Intelligence* 49 (1991) 61-95
6. Dechter, R., Pearl, J.: Network-based heuristics for constraint-satisfaction problems. *Artificial Intelligence*, 34 (1987) 1-38
7. Dubois, D., Prade, H.: *Possibility Theory: An approach to computerized processing of uncertainty*. Plenum Press, New York (1988)
8. Even, S.: *Graph Algorithms*. Computer Science Press, Rockville, MD (1979)
9. Freuder, E.: A sufficient condition for backtrack-free search. *Journal of the ACM* 29, 1 (1982) 24-32
10. Kondrak, G., van Beek, P.: A Theoretical Evaluation of Selected Backtracking Algorithms. *Artificial Intelligence* 89 (1997) 365-387
11. Mackworth, A.: Consistency in networks of relations. *Artificial Intelligence* 8 (1977) 99-118
12. Marín, R., Barro, S., Bosch, A., Mira, J.: Modelling the representation of time from a fuzzy perspective. *Cybernetics and Systems*, 25, 2 (1994) 207-215
13. Marín, R., Barro, S., Palacios, F., Ruiz, R., Martín, F.: An Approach to Fuzzy Temporal Reasoning in Medicine. *Mathware & Soft Computing* 3 (1994) 265-276
14. Marín, R., Cardenas, M., Balsa, M., Sanchez, J.: Obtaining solutions in fuzzy constraint networks. *Int. Journal of Approximate Reasoning*, 16 (1997) 261-288
15. Meiri, I., Dechter, R., Pearl, J.: Uncovering trees in constraint networks. *Artificial Intelligence*, 86 (1996) 45-267
16. U. Montanari, *Networks of constraints: fundamental properties and applications to picture processing*. *Inf. Sci.*, 7 (1974) 95-132.
17. Navarrete, I., Marín, R., Balsa, M.: *Redes de Restricciones Temporales Disyuntivas Borrosas*. ESTYLF'95, Murcia, Spain (1995) 57-63
18. Schwalb, E., Dechter, R.: Coping With Disjunctions on Temporal Constraint Networks. *Proc. AAAI'93*, Washington (1993) 127-132
19. Schwalb, E., Dechter, R.: Processing Disjunctions in Temporal Constraint Networks. *Artificial Intelligence* 93 (1997) 29-61
20. Schwalb, E., Vila, L.: Temporal Constraints: A Survey. *Constraints* 3(2/3) (1998) 129-149
21. Stergiou, K., Koubarakis, M.: Backtracking Algorithms for Disjunctions of Temporal Constraints. *Artificial Intelligence* 120 (2000) 81-117
22. Tsang, E. *Foundations of Constraint Satisfaction*. Academic Press, London (1993)
23. Wald, J.A., Colburn, C.J.: Steiner Trees, Partial 2-Trees and Minimum IFI Networks. *Networks* 13 (1983) 159-167