A flexible temporal planner *

Luis Castillo, Juan Fdez-Olivares, Antonio González

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S. Ingeniería Informática. Universidad de Granada, 18071 Granada, SPAIN.
{L.Castillo,faro,A.Gonzalez}@decsai.ugr.es

Abstract. The idea of including a flexible management of time in a temporal planner is the aim of this work. We propose an extension of a previous work of the authors [4] based on the use of fuzzy temporal intervals by defining a possibility measure over the continuous time line. This gives the planner some very interesting features like obtaining valid plans when the knowledge is ill defined, obtaining approximate plans when exact plans are not possible and provide a flexible time line of execution.

1 Introduction

Temporal planners [1, 9, 14–16, 18] use a rather rigid notion of time in the sense that time and durations are assigned and compared taking into account only strict equality. This leads to the design of temporal plans that must be executed exactly in the time line defined by the planner since, otherwise, the plan will not work. In realistic problems this is a very restrictive notion of time since many of the time points or durations implicitly computed in a plan may be ill defined, due to incomplete or vague knowledge, unpredictable behaviors or execution errors. In these cases, temporal plans could be more flexible in the handling of time, in the sense that they might be able to handle ill defined temporal knowledge or some variations along the time line of the execution of the plan in order to cope with these uncertainties. For example, when one makes a plan in order to go to work from home, one doesn't know the exact time that the bus arrives at the bus stop, nor the exact time that the bus takes to complete the route. Therefore, in order to succeed in the execution of this plan, one predicts how long in advance one must leave the home to take the bus. None of these times are precisely known but plans still succeed since people continue using the bus.

This paper is an extension of a previous work of the authors which defined a Temporal Constraint Network (TCN) based temporal planner able to introduce some uncertainty in a temporal plan by using bounded temporal constraints thanks to the expressiveness of TCNs [4, 8]. Now, in this paper, these temporal intervals are transformed into fuzzy temporal intervals by defining a possibility measure over them in order to recognize that not every time point in a temporal

* This work has been supported by the Spanish MCyT under projects TAP99-0535-C02-1 and TIC2001-4936-E.

interval is equally *preferred* or *certain* and that, in many realistic applications, constraints are more or less relaxed in practice. This allows the planner to build plans based on vague temporal constraints of the form: "the action of filling the tank will take about 100 time units" or "action a must be delayed more or less between 20 and 30 time units after action b". This type of "soft constraints" provide several advantages:

- It is possible to model domains in which temporal knowledge is vaguely known and to obtain temporal plans of practical use for these domains.
- It becomes a very interesting approach in time critical problems, like space missions or complex scheduling problems, where execution times are very tight and perhaps a solution that completely meet the temporal constraints is not possible and, instead, an approximate solution may be found.
- Soft temporal constraints are also very important in the framework of Fuzzy Temporal Constraint Networks. A temporal plan built on top of a FTCN is not a solution, but a class of possible solutions, i.e., slightly different temporal schedules of the same set of actions that may be obtained by a solution extraction procedure [8, 17]. This means that a temporal plan may be seen under a very flexible point of view. Once a solution has been extracted and it has started its execution, if an unexpected delay occurs, it is still possible to obtain a new time line for the remaining of the plan without the need to replan, just by adapting the plan to the existing delay.

Soft constraints have been successfully used to model and solve different realistic domains as for example in industrial optimization problems [5], robotic vehicle routing [6], electrical distribution problems [7], patient supervision in intensive care units [2] or industrial job-shop scheduling [11]. Fuzzy temporal constraints have also been investigated and there are different papers [17, 19] in the literature about FTCN. But most of them are somehow "a posteriori" approaches, i.e., they are a further work on a FTCN previously built upon some evidences or observations like in diagnosis or monitoring applications. This paper introduces a new perspective in FTCN since temporal planning needs an "a priori" approach to handle these models, i.e., while several different, alternative, FTCN are being built to solve the same problem, the planner has to choose the best one in order to meet the goal of the planning problem and now the question is: which FTCN, amongst every possible FTCN for the same temporal problem, provide better temporal plans?

2 An overview of MACHINE^T

MACHINE^T is a partial order, causal-link based temporal planner [4]. Domains are described as a set of agents and every agent is described by means of a Finite State Automaton (FSA) as shown in Figure 1.

Thus, every agent has a set of internal states and a set of actions, each of whom produces a change of state (immediate, since they are internal states) and a set of effects in the environment (every effect takes some time to be achieved

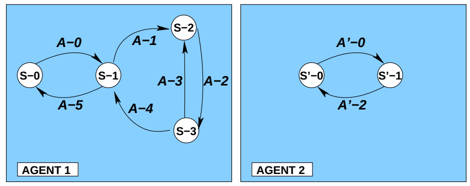


Fig. 1. The structure of a domain of MACHINE^T which shows two agents and several actions available for each agent. Circles represent states and arcs represent actions.

and during this delay the effect is undefined). Figure 2 shows the structure of the domain of the zenotravel problems [13] in terms of this FSA representation.

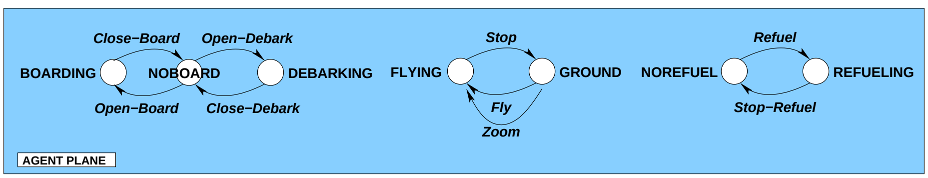


Fig. 2. The structure of the well known zeno travel problem expressed in MACHINE^T by means of a complex FSA

Given this representation, durative actions are represented taking into account the Newtonian assumption that once an action has started its execution, it will continue executing until some other event interrupts it. Therefore, once an action has achieved a change of state in its agent, it will continue until the next change of state of the agent or until the end of the plan. For example, given the domain of the zenotravel problem in Figure 2 action `Fly` achieves the state `FLYING` and it will continue executing until the next change of state, i.e., until action Stop or the end of the plan, so its possible intervals of execution are either `[Fly, Stop]` or `[Fly, End-Of-Plan]` depending on how the plan has been designed.

In addition, invariant conditions of PDDL 2.1 level 3 [12] are also easily represented associated to these intervals. For example, following the zeno example, action `Fly` requires that before and during its execution the plane must have finished any possible refueling operation (i.e. `(STATE PLANE NO-REFUEL)`). This is represented as a simultaneous requirement of action `Fly` and protected by means of a causal link [4, 21] along its interval of execution, that is, either `[Fly, Stop]` or `[Fly, End-Of-Plan]`.

MACHINE^T also allows to represent both deadline goals and makespans of plans. Deadline goals are goals that must be achieved exactly in a given period of time $[t_{min}, t_{max}]$. This model does not allow slacks during the satisfaction of deadline goals. Makespan of plans is the expected duration of the plan and it is also specified in the form of an interval of time points $[t_{min}, t_{max}]$ standing for the minimum and maximum duration of the plan (see [4] for details).

Temporal plans are built on top of a simple Temporal Constraint Network (TCN) [8] where every action is represented as a node of the TCN and temporal constraints are posted during the resolution process every time that an action solves a subgoal or when actions are reordered to avoid interferences between them. The result is a partially ordered plan (see Figure 3) whose actions are annotated with an earliest and a latest execution time (taking into account that the beginning of the plan has an absolute time point zero). These temporal plans do not represent a temporal solution to the problem being solved, but a class of solutions where the exact execution time for every action is obtained by means of a very simple solution extraction procedure [8].

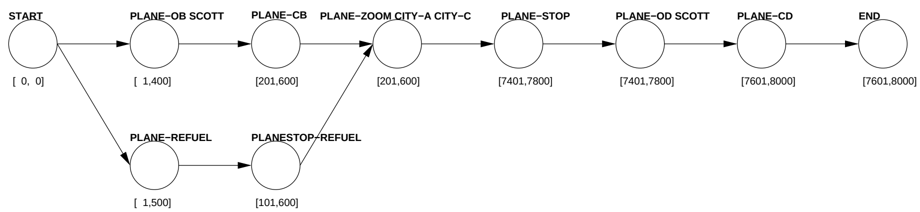


Fig. 3. A temporal plan showing the earliest and latest execution times for every action. The plan was built taking into account a possible makespan of $[0, 8000]$ time units.

3 The introduction of fuzzy notion of time as a more rational temporal uncertainty

The uncertainty shown in temporal plans, mainly in execution times (see Figure 3) but also in any relative constraint between actions, comes from the following assumptions. Firstly, every effect of an action may not be instantaneous but it might have a delay. For example, action `Refuel Plane` in the domain of Figure 2 takes 100 time units to refuel a plane and action `Open Boarding ?person` takes 200 time units to board a person.

And secondly, once all of the preconditions of a consumer action have been achieved at time t the consumer action might be executed at time t but however, *it might also be executed at some arbitrary time point* greater than or equal to t whenever it did not interfere with the rest of the plan and in order to fit more general or overall temporal constraints of the problem. For example, if the preconditions `Refueled Plane` and `Boarded ?person` of action `Fly` have been achieved at time units 100 and 200 respectively, then action `Fly` may start at time unit 200 but it might be delayed as long as needed. Therefore the time of achievements of preconditions is only a minimum time for the execution of the consumer action. This implicitly leads to the posting of constraints of the form

$[t_{min}, +\infty)$ between producer and consumer actions that become closed intervals of the form $[t_{min}, t_{max}]$ only when deadline goals or makespans are present in a problem.

The representation and handling of these uncertain temporal references is a very important issue in realistic problems where unexpected faults and delays may occur during the execution of the plan. In these cases, once an initial solution has been extracted from the temporal plan and it has started its execution, the occurrence of a delay in the execution of an action may be still acceptable if the detected delay fits in the set of constraints of the temporal plan, otherwise the delay will be inconsistent with the plan and a new plan should be built.

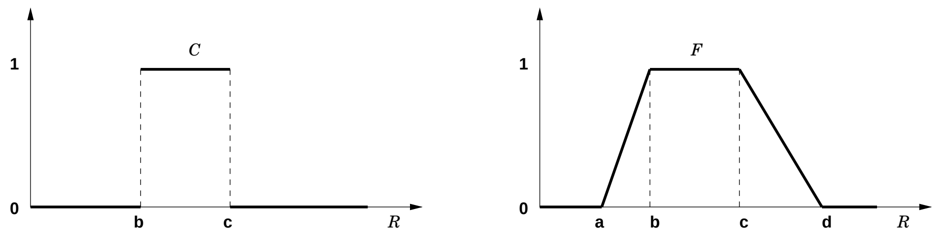
However, although this capability is very useful for realistic problem solving and monitoring, the use of intervals may not be as rational as desired since they suddenly change from consistency to inconsistency in the boundary of the interval. Let consider the following example. A plan has been built for the zeno problem and a flight has been scheduled to take no more than 8000 time units, but due to an unexpected delay in the execution of an action, very useful in the aerospace domain indeed, a new solution can be obtained but it is 8005 time units long. Is it acceptable or not? The answer could be that it is not as acceptable as a solution in the correct bounds, but if no such a solution may be found, then it may be considered rather acceptable.

This leads to the acknowledgement of relaxed constraints and different degrees of acceptability for a set of solutions or, in terms of temporal reasoning, to different degrees of temporal consistency. The remaining of the paper describes the use of fuzzy sets theory to represent and handle temporal plans that accept solutions with different degrees of consistency thanks to the use of Fuzzy Temporal Constraint Networks [17, 19].

3.1 A brief introduction to fuzzy sets

A fuzzy set, or more precisely a fuzzy subset, is a subset of a reference set whose boundaries are not precisely known [22]. The left hand part of Figure 4 shows a classic, or crisp, subset C whose membership is given by the function π_C which hardly restricts the membership to the subset. The right hand part of Figure 4 shows a fuzzy subset F whose membership is given by the possibility distribution π_F. This is a function from the reference set R to the real interval $[0, 1]$ which softly restricts the membership to the subset. In this case $\pi_F(x)$ is a numerical estimation for the fact that element x can be considered as a member of the subset F.

Taking into account an ill defined subset or relation, this possibility distribution may be used to model either uncertainty, not related to probability, or preference with respect to the elements of the reference set [10, 22]. The greater is the difference between values a and b and values c and d, the fuzzier is the subset and it may be used by a modeler to represent the uncertainty about the membership to the fuzzy subset or the degree of preference with respect to the values of the reference set.



$$\forall x \in R,\ \pi_C(x) = \begin{cases} 0 & \text{if } x < b \\ 1 & \text{if } b \leq x \leq c \\ 0 & \text{if } c < x \end{cases} \qquad \forall x \in R,\ \pi_F(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x < b \\ 1 & \text{if } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{if } c < x \leq d \\ 0 & \text{if } d < x \end{cases}$$

Fig. 4. A classic, or crisp, subset (left) C and a fuzzy subset (right) F of the reference set R between points b and c and their respective membership functions.

A fuzzy subset F, or its possibility distribution π_F, is said to be unimodal if it defines a convex subset, i.e., $\forall x_1, x_2, x_3 \in R$ such that $x_3 \in [x_1, x_2]$ it holds that $\pi_F(x_3) \geq min\,\{\pi_F(x_1), \pi_F(x_2)\}$.

A fuzzy subset F, or its possibility distribution π_F, is said to be normalized if $\exists x \in R$ such that $\pi_F(x) = 1$. The height of a fuzzy subset is $hgt(F) = \max_{x \in R} \pi_F(x)$ that, in the case of normalized fuzzy subsets is equal to 1.

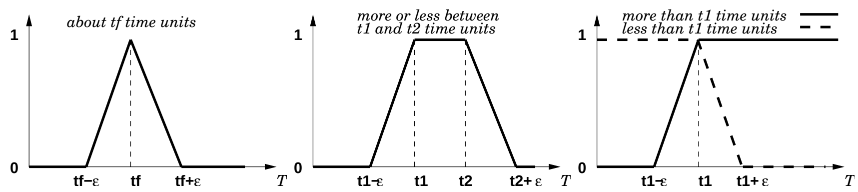
In the following, only unimodal, not necessarily normalized, fuzzy sets are considered.

3.2 Representation of fuzzy time in Machine^{TF}

A fuzzy date D is an imprecise reference to a time point or to a temporal interval. It may be represented as a fuzzy subset of a continuous time scale T so that $\pi_D(t), t \in T$ represents the degree of possibility for the fuzzy date D to take the exact value t. This concept of fuzzy date has been used to design a flexible temporal planner called Machine^{TF} on the basis of the the temporal features of a crisp temporal planner explained in [4]. These features are summarized in the following.

- Delays of effects. Every effect f of every action takes about t_f time units to be achieved (see Figure 5.a). This temporal reference is noted as $\widetilde{t_f}$ and its fuzziness is decided by the domain designer according to the uncertainty of the problem.
- Execution times. Some actions have not a predefined execution time but others have a fixed execution time of $\widetilde{t_f}$ time units (see Figure 5.a).
- Deadline goals. There may be goals which should be achieved more or less between $t1$ and $t2$ time units (see Figure 5.b). This is noted as $[\widetilde{t1}, \widetilde{t2}]$.

– Plan makespan. A plan may be required to be more or less between $t1$ and $t2$ time units long (see Figure 5.b).
– Reordering of actions. In order to prevent harmful interactions between mutually interfering actions, an action a may be reordered more (less) than t time units after (before) action b (see Figure 5.c). This is noted as $[\widetilde{t}, +\infty)$ or $(-\infty, \widetilde{t}]$.



(a) (b) (c)

Fig. 5. The fuzzy temporal references that are used in MACHINE^{TF}. The value ϵ may be used by a modeler to increase or decrease the fuzziness of the subset

These temporal features, together with their fuzzy representations, are handled by MACHINE^{TF}, as will be explained later in Section 5, to build temporal plans of top of a fuzzy temporal constrain network. These plans are expected to solve temporal problems where time is not precisely known and to provide a flexible time line for its execution.

3.3 Temporal plans as fuzzy temporal constraint networks

Definition 1. *A Fuzzy Temporal Constraint Network* $\mathcal{N} = \langle \mathcal{X}, \mathcal{C} \rangle$ *is composed of a set of variables* $\mathcal{X} = \{X_0, X_1, \ldots, X_{n+1}\}$ *and a set of fuzzy binary temporal constraints defined between them* $\mathcal{C} = \{C_{ij} | \ 0 \leq i, j \leq n+1\}$ *[17, 19].*

Every fuzzy binary constraint C_{ij} restricts the possible relative values of X_i and X_j, i.e., $X_j - X_i \leq C_{ij}$. Every constraint C_{ij} is defined as a fuzzy date as explained in the previous section and represented by a possibility distribution π_{ij} over the continuous time scale T. Every variable X_i is a crisp variable that takes values of the time scale T restricted to the former fuzzy temporal constraints. Variables X_0 and X_{n+1} are two dummy variables used to represent the beginning and the end of the network.

A solution to a FTCN is an assignment of the form $X_i = x_i, x_i \in T$. Every fuzzy temporal constraint of the form C_{0i} restricts the absolute possible values of variable X_i and constraints of the form C_{ij} $i, j > 0$ restricts the possible relative values of X_i and X_j. Then, a FTCN do not have a unique solution, but a set of solutions each of whom has a degree of consistency (a degree of possibility) given by the individual degrees of consistency of every single assignment with respect to their respective fuzzy temporal constraint.

Definition 2. *A σ-possible solution of a FTCN $\mathcal{N}$ is a tuple $s = (x_0, x_1, \ldots, x_{n+1})$ that verifies that $\pi_S(s) = \sigma$ and*

$$\pi_S(s) \;=\; \min_{0 \le i,j \le n+1} \pi_{ij}(x_j \;-\; x_i) \tag{1}$$

Then, the set of solutions that might be obtained for a FTCN is featured by the following value.

Definition 3. *A FTCN is α-consistent if the set of possible solutions $S \subseteq \mathcal{R}^{n+2}$ verifies*

$$\sup_{s \in S} \pi_S(s) = \alpha \tag{2}$$

Given a FTCN, there are some different ways to extract its solutions, at different degrees of consistency, and they may be consulted in [17]. In the framework of MACHINE^{TF}, a minimal FTCN, that is, the transitive closure of a FTCN [8, 17, 19], is used to represent a fuzzy temporal plan where every variable represents the execution time of one of the actions of the plan and every fuzzy binary temporal constraint is posted during the resolution process as will be explained later. For example, a fuzzy temporal plan similar to that of Figure 3 is shown in Figure 6.

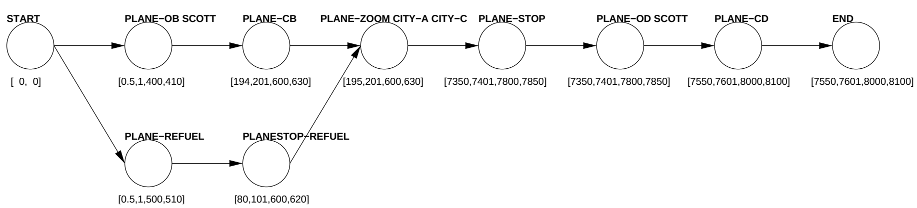


Fig. 6. A fuzzy temporal plan showing the fuzzy temporal reference (π_{0i}) that constraints the possible values of the execution time of each action. Every fuzzy temporal reference is expressed by a 4-tuple $[a, b, c, d]$ as in the fuzzy subset of the right hand side of Figure 4

The capability to build these fuzzy temporal plans provides the following advantages. Firstly,they may be used to solve problems with ill defined temporal knowledge since classic temporal plans cannot be used due to their rigid interpretation of time. Secondly, in hard temporal problems, they may be used to obtain approximate solutions where absolute solutions are not possible thanks to a flexible interpretation of temporal constraints. And thirdly, they provide a flexible time line for their execution able to adapt the execution of the plan to unexpected delays without the need to replan. Even more, this model might also be used to solve classic temporal problems where time is perfectly known just by replacing their corresponding possibility distribution by a crisp one: $\pi_t(t') = 1$ if $t' = t, 0$ otherwise.

However the use of FTCN in a temporal planning framework poses some interesting questions that are outlined in the next section.

4 Heuristic featuring of fuzzy temporal constraint networks

The literature about FTCNs is devoted to further works or refinements over previously built FTCNs, given some previously known evidences and knowledge, like in deductive and reasoning processes, or in diagnosis and monitoring applications [2, 20]. Contrary to these approaches, the inclusion of FTCN in a planning framework needs an "a priori" study of FTCNs since they are part of a greater search process in which they are being built to support fuzzy temporal plans. Let see it with the following example.

Consider again the zeno travel domain whose structure has been shown in Figure 2. Consider as well that a given makespan has been given so that the use of the faster action `Zoom` allows to obtain a FTCN with $\alpha = 1$, that is, completely consistent with the problem and that, instead, the use of the slower action `Fly` only allows to obtain solutions with $\alpha = 0.5$ or lower. The question arises when a choice has to be made on both plans in order to decide which one is the best solution. In this case, both have the same capability for solving the problem so a complementary criterion is needed. Then the degree of consistency α plays a very important role. In this case, the plan with the action `Zoom` and $\alpha = 1$ should be more preferred, since it allows to obtain completely consistent solutions, with respect to the other plan, with the action `Fly` and $\alpha = 0.5$, only able to obtain approximate solutions.

This answer may seem easy but it poses a difficult question. Given Equation 2, the value of the maximum degree of consistency α of a FTCN might be a very good source of information for decision making, but it seems difficult to obtain analytically. Perhaps some search algorithm like simulated annealing or genetic algorithms could be used to guess that value, but this will degrade severely the performance of MACHINE^{TF} since this process should be launched to evaluate every possible plan being built.

In the framework of MACHINE^{TF}, although the value of α is not known, it may be bounded and used to guide the search of the planning process.

A Montecarlo algorithm is used several times to randomly generate a set of candidate solutions. Infeasible solutions are rejected and the remaining ones are evaluated following Equation 1 such that the best degree of possibility of these candidates is kept $\delta_l = \underset{i}{max}\ \pi_S(sol_i)\ i = 1, \ldots, k$ [1]. Clearly, $\delta_l \leq \alpha$.

When all of the fuzzy temporal constraints in a minimal FTCN are normalized, i.e., $hgt(\pi_{ij}) = 1\ \forall i, j$ the degree of consistency α is equal to 1, that is, there is at least one solution that completely meets the constraints [17]. Otherwise, let us consider the following value $\delta_u = \underset{i}{min}\ hgt(\pi_{0i})$. Then, given equation 2, it is easy to prove that $\alpha \leq \delta_u$. This is a rather good estimation since it has been obtained from a simplified model. In the case of absolute FTCNs, that is, FTCNs where only absolute constraints of the form C_{0i} are posted and every relative constraint $C_{ij}, i, j \neq 0$ is a transitive constraint, it may be proven that $\alpha = \delta_u$.

[1] The number or runs, k, is fixed previously to the planning process.

Then, both values δ_l and δ_u bounds the unknown value of α

$$\alpha \in [\delta_l, \delta_u]$$

Since the exact value of α is unknown, the centroid of this interval is used to estimate the "goodness" of the solutions provided by a FTCN. This value is then used by MACHINE^{TF} as a secondary ranking criterion during the search process, where the primary ranking criterion is the heuristic evaluation function of MACHINE, that has proven to be very powerful in several domains [3].

5 MACHINE^{TF}: A flexible temporal planner

The algorithm of MACHINE^{TF} is basically, a partial order, causal link based planning algorithm [21] and it is exactly the same than in MACHINE^{T} [4], but taking into account the introduction of fuzzy temporal constraints:

Goal Satisfaction

- Every time an action a with a delayed effect, as a temporally annotated literal $(f, \widetilde{t_f})$, solves a precondition of an action b, a fuzzy temporal constraint of the form $[\widetilde{t_f}, +\infty)$ (see Section 3.2) is posted from b to a.
- Every action a with a fixed duration $\widetilde{t_f}$ adds a new fuzzy temporal constraint of the form "about t_f time units" between the action a and the end of its interval of execution (see Section 2). In the case that the action does not have a fixed duration, then the constraint is
$$\bigwedge_{f' \ in \ effects(a)} [\widetilde{t_{f'}}, +\infty)$$
that is, at least the maximum delay time of effects.
- For every deadline goal, as a temporally annotated goal $(g, [\widetilde{t_{min}}, \widetilde{t_{max}}])$, meaning that goal g must be achieved more or less between times $\widetilde{t_{min}}$ and $\widetilde{t_{max}}$, which has been solved by an action a with a temporally annotated effect $(f, \widetilde{t_f})$ a constraint is added of the form $[\widetilde{t_{min}} \ominus \widetilde{t_f}, \widetilde{t_{max}} \ominus \widetilde{t_f}]$ (see Section 3.2) between the time points of action $START$ and a, given that time point of action $START$ has an absolute time zero[2].

Concurrent actions and threats Say that action $a3$ with the delayed effect $((not\ f), \widetilde{t_f})$ interferes with a casual link from actions $a1$ to $a2$ with respect to the effect f. Then promotion adds the constraint
$$\bigwedge_{f' \ in \ effects(a2)} [\widetilde{t_{f'}}, +\infty)$$
between $a2$ and $a3$. Demotion adds the constraint $[\widetilde{t_f}, +\infty)$ from action $a3$ to action $a1$.

Makespans Makespans for the fuzzy temporal plan can be very easily represented by adding a fuzzy temporal constraint of the form $[\widetilde{t_{min}}, \widetilde{t_{max}}]$ between actions $START$ and END at the beginning of the planning process.

[2] Where the operation $\ominus$ is the subtraction operation defined for fuzzy sets [10, 17].

In these terms, MACHINE^{TF} is a temporal planner able to represent ill defined temporal knowledge and to obtain fuzzy temporal plans expressed as a plan distributed along a minimal FTCN. These temporal plans are not a solution to the temporal problem being solved, but a set of solutions, so one of the methods proposed in [17] is used[3] to obtain one of these solutions and its corresponding degree of possibility.

6 Conclusions

This paper has outlined MACHINE^{TF}, a temporal planner based on the representation of fuzzy temporal knowledge. The main contributions of MACHINE^{TF} are the following.

- It is able to obtain plans in domains in which time is not precisely known. This is very important in real world problems and mainly with respect to classic temporal planners, which need a rigid representation of time and, therefore, they lose most of the temporal expressiveness of these domains.
- It is also able to obtain approximate solutions where exact solutions are not feasible, given a soft interpretation of temporal constraints. This is also very important in hard temporal problems since it transforms the inconsistency in a matter of preference and degrees, more in accordance with most usual constraints in real life.
- And finally, fuzzy temporal plans obtained provide a flexible time line for its execution. Given that a fuzzy temporal plan may be scheduled in different time lines, depending on the solution that has been extracted, if an unexpected delay occurs (due to an error, a delay in the execution of an action, etc), a new solution adapted to the new situation may be re-extracted without the need to replan.

References

1. F. Bacchus and M. Ady. Planning with resources and concurrency. a forward chaining approach. In *IJCAI'01*, 2001.
2. S. Barro, R. Marín, F. Palacios, and R. Ruiz. Fuzzy logic in patient supervision systems. *Artificial Intelligence in Medicine*, 21:193–199, 2001.
3. L. Castillo, J. Fdez-Olivares, and A. González. Mixing expresiveness and efficiency in a manufacturing planner. *Journal of Experimental and Theoretical Artificial Intelligence*, 13:141–162, 2001.
4. L. Castillo, J. Fdez-Olivares, and A. González. A temporal planner based on temporal constraint networks to solve real-life problems. *Submitted to Applied Artificial Intelligence*, 2002. Available as DECSAI Technical report DECSAI-020301.
5. L. Castillo and A. González. Fuzzy optimization of distribution networks by using genetic algorithms. In F. Herrera and J.L. Verdegay, editors, *Genetic Algorithms and Soft Computing*, Studies in Fuzziness and Soft Computing, pages 641–657. Springer-Verlag, 1996.

[3] We have used Method 3 [17] because of its low computational cost.

6. R. Cheng and M. Gen. Fuzzy vehicle routing and scheduling problem using genetic algorithms. In F. Herrera and J.L. Verdegay, editors, *Genetic Algorithms and Soft Computing*, Studies in Fuzziness and Soft Computing, pages 683–709. Springer-Verlag, 1996.
7. O. Cordón, F. Herrera, and L. Sánchez. Solving electrical distribution problems using hybrid evolutionary data analysis techniques. *Applied Intelligence*, 10:5–24, 1999.
8. R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.
9. M. Do and S. Kambhampati. SAPA: a domain-independent heuristic metric temporal planner. In *European Conference on Planning*, pages 109–120, 2001.
10. D. Dubois and H. Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York, 1988.
11. Didier Dubois, Hélène Fargier, and Henri Prade. The use of fuzzy constraints in job-shop scheduling. In *Proc. of IJCAI-93/SIGMAN Workshop on Knowledge-based Production Planning, Scheduling and Control*, Chambery, France, August 1993.
12. M. Fox and D. Long. PDDL2-1: an extension to PDDL for expressing temporal planning domains. Technical report, University of Durham, UK, 2001.
13. M. Fox and D. Long. Domains of the 3rd. international planning competition. In *Artificial Intelligence Planning Systems (AIPS 02)*, 2002.
14. A. Garrido, E. Onaindía, and F. Barber. Time-optimal planning in temporal problems. In *European Conference on Planning*, pages 397–502, 2001.
15. P. Haslum and H. Geffner. Heuristic planning with time and resources. In *European Conference on Planning*, pages 121–132, 2001.
16. P. Laborie and M. Ghallab. Planning with sharable resource constraints. In *IJCAI'95*, pages 1643–1649, 1995.
17. R. Marín, M.A. Cárdenas, M. Balsa, and J.L. Sánchez. Obtaining solutions in fuzzy constraint networks. *International Journal of Approximate Reasoning*, 16:261–288, 1997.
18. D.E. Smith and D.S. Weld. Temporal planning with mutual exclusion reasoning. In *IJCAI'99*, pages 326–337, 1999.
19. L. Vila and L. Godo. On fuzzy temporal constraint networks. *Mathware and Soft Computing*, 3:315–334, 1994.
20. L. Vila and L. Godo. Query answering in fuzzy temporal constraint networks. In *IJCAI'95: Proceedings of the Intl. Joint Conference on Atificial Intelligence*, volume 3, pages 315–334, 1994.
21. D. Weld. An introduction to least commitment planning. *AI Magazine*, 15(4):27–61, 1994.
22. L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.