Spatial Cognition & Computation, 8:131–149, 2008

ISSN: 1387-5868 print/1542-7633 online
DOI: 10.1080/13875860801930407

Managing Temporal Constraints with Preferences

Malek Mouhoub and Amrudee Sukpan
University of Regina

Abstract: Preferences in temporal problems are common but significant in many real world applications. In this paper, we extend our temporal reasoning framework, managing numeric and symbolic information, in order to handle preferences. Unlike the existing models managing single temporal preferences, ours supports four types of preferences, namely: *numeric* and *symbolic temporal preferences, composite preferences* and *conditional preferences*. This offers more expressive power in representing a wide variety of temporal constraint problems. The preferences are considered here as a set of soft constraints using a *c-semiring* structure with combination and projection operators. Solving temporal constraint problems with preferences consists in finding a solution satisfying all the temporal constraints while optimizing the preference values. This is handled by a variant of the branch and bound algorithm, we propose in this paper, and where constraint propagation is used to improve the time efficiency. Experimental tests, we conducted on randomly generated temporal constraint problems with preferences, favor a variant of MAC as the constraint propagation strategy that should be used within the branch and bound algorithm.

Keywords: temporal reasoning, constraint satisfaction, preference reasoning

1. INTRODUCTION

Temporal preferences play an important role in many real world applications. In general, they express noncrisp desire of start/end times, time intervals, and temporal relations of feasible scenarios. Obviously, preferences are not hard constraints that have to be fully satisfied, but have an effect on choosing a good or the best scenario satisfying all the hard constraints. Moreover, often temporal preferences are implicit. In order to deal with temporal preferences such as *early, late, about 6 pm, etc.*, we need to transform each of them into a formal explicit preference function. Furthermore, these preference functions are often combined with other forms of preferences in order to have a global preference for a given temporal scenario.

Received April 16, 2007; Accepted November 30, 2007
Correspondence concerning this article should be addressed to Malek Mouhoub, Dept. of Computer Science, University of Regina, 3737 Wascana Pkwy, Regina SK S4S0A2 Canada. E-mail: mouhoubm@cs.uregina.ca

In Mouhoub and Sukpan (2004) we have proposed a modeling framework that allows the management of numeric and symbolic time information within a unique constraint network. In addition, this model enables the addition of temporal information dynamically to the problem to solve, during the resolution process, via composite variables and activity constraints. Composite variables are variables whose possible values are temporal events.[1] In other words this allows us to represent disjunctive temporal events. An activity constraint has the following form $X_1 \wedge \ldots X_p \overset{condition}{\rightarrow} Y$ where $X_1, \ldots, X_p$ and Y are temporal variables (composite or events). This activity constraint will activate Y (Y will be added to the problem to solve) if $X_1 \wedge \ldots X_p$ are active (currently present in the problem to solve) and *condition* holds between these variables. *condition* corresponds to the assignment of particular values to the variables $X_1, \ldots, X_p$. We call Conditional and Composite Temporal Constraint Satisfaction Problem (CCTCSP) this model we have proposed.

In this paper, the CCTCSP is extended to include four types of temporal preference: *numeric*, *symbolic, composite*, and *conditional preferences*. We call this model CCTCSP with Preferences (or CCTCSPP). *Numeric* and *symbolic temporal preferences* associate degrees of preferences respectively to time intervals and symbolic relations, in order to favor some temporal decisions. A *composite preference* is a higher level of preference among the temporal choices of a composite variable. *Conditional preferences* allow some preference functions (numeric, symbolic or composite) to be added dynamically to the problem (associated to a given event or composite variable), during the resolution process, if a given condition on some temporal variables is true. Solving a CCTCSP is a decision problem which consists in finding an assignment of time intervals to the temporal events such that all the constraints are satisfied. This can be handled by approximation methods based on stochastic local search or by a systematic backtrack search algorithm where constraint propagation is used to prevent earlier later failure (Mouhoub & Sukpan, 2005a, 2005b). On the other hand, solving a CCTCSPP is an optimization problem which consists of finding the best solution according to the preference values. This can be done by a variant of the branch an bound algorithm we propose in this paper. Note that constraint propagation is also used here to prune some inconsistent values at the early stage of the resolution process. Experimental tests we conducted on randomly generated temporal constraint problems with preferences, favor a variant of MAC as the constraint propagation strategy to use within the branch and bound algorithm.

In the next section we will introduce the CCTCSP model and its related solving techniques. In Section 3 we will summarize the related work in the area of temporal preferences. Section 4 is then dedicated to numeric, symbolic, composite, and conditional preferences. In Section 5 we present the branch and bound algorithm for solving CCTCSPPs. Section 6 is dedicated

[1] An event is defined here as a proposition that holds over a time interval.

to the experimental tests we conducted on randomly generated CCTCSPPs. Conclusion and perspectives are finally listed in Section 7.

2. MANAGING CONDITIONAL CONSTRAINTS AND COMPOSITE VARIABLES

In the following, we define the CCTCSP model and its corresponding constraint network (graph representation) through an example.

Definition 1: Conditional and Composite Temporal Constraint Satisfaction Problem (CCTCSP). A Conditional and Composite Temporal Constraint Satisfaction Problem (CCTCSP) is a tuple $\langle E, D_E, X, D_X, IV, C, Act\rangle$, where

$E = \{e_1, \ldots, e_n\}$ is a finite set of temporal variables that we call events. Events have a uniform reified representation made up of a proposition and its temporal qualification: $Evt = OCCUR(p, I)$ defined by Allen (1983) and denoting the fact that the proposition p occurred over the interval I. For the sake of notation simplicity, an event is used in this paper to denote its temporal qualification.

$D_E = \{D_{e_1}, \ldots, D_{e_n}\}$ is the set of domains of the events. Each domain D_{e_i} is the finite and discrete set of numeric time intervals the event e_i can take. D_{e_i} is expressed by the 4-tuple [$begintime_{e_i}$, $endtime_{e_i}$, $duration_{e_i}$, $step_{e_i}$] where $begintime_{e_i}$ and $endtime_{e_i}$ are respectively the earliest start time and the latest end time of the corresponding event, $duration_{e_i}$ is the duration of the event and $step_{e_i}$ defines the distance (number of time units) between the starting time of two adjacent intervals within the event domain. The discretization step $step_{e_i}$ allows us to handle temporal information with different granularities.

$X = \{x_1, \ldots, x_m\}$ is the finite set of composite variables.

$D_X = \{D_{x_1}, \ldots, D_{x_m}\}$ is the set of domains of the composite variables. Each domain D_{x_i} is the set of possible events the composite variable x_i can take.

IV is the set of initial variables. An initial variable can be a composite variable or an event. $IV \subseteq E \bigcup X$.

$C = \{C_1, \ldots, C_p\}$ is the set of *compatibility constraints*. Each compatibility constraint is a qualitative temporal relation between two variables in case the two variables are events, or a set of qualitative relations if at least one of the two variables involved is composite. A qualitative temporal relation is a disjunction of Allen primitives (Allen, 1983).

Act is the set of *activity constraints*. Each activity constraint has the following form:

$X_1 \wedge \ldots X_p \overset{condition}{\rightarrow} Y$ where $X_1, \ldots, X_p$ and Y are temporal variables (composite or events). This activity constraint will activate Y if $X_1, \ldots, X_p$ are active and *condition* holds on these variables. *condition* corresponds to the assignment of particular values to the variables $X_1, \ldots, X_p$.

Let us illustrate the CCTCSP through the following example.

Example 1. *John, Mike and Lisa are going to see a movie on Friday. John will pick Lisa up and Mike will meet them at the theater. If John arrives at Lisa's before 7:30, then they will stop at a convenient store to get some snacks and pops. It will take them 30 minutes to reach the theater if they stop at the store and 15 minutes otherwise. There are three different shows playing: movie*$_1$*, movie*$_2$*, and movie*$_3$*. If they finish the movie by 9:15, they will stop at a Pizza place 10 minutes after the end of the movie and will stay there for 30 minutes. John leaves home between 7:00 and 7:20. Lisa lives far from John (15 minutes driving). Mike leaves home between 7:15 and 7:20 and it takes him 20 minutes to go to the theater. movie*$_1$*, movie*$_2$*, and movie*$_3$ *start at 7:30, 7:45, and 7:55 and finish at 9:00, 9:10, and 9:20, respectively.*

The goal here is to check if this story is consistent (has a feasible scenario). The story can be represented by the CCTCSP in Figure 1. Each event domain is represented by the 4-tuple [*begintime*, *endtime*, *duration*, *step*]. In the case of *John_Pick_Lisa*, the domain is $[0, 35, 15, 1]$ where 0 (the time origin corresponding to 7:00) is the earliest start time, 35 is the latest end time, 15 is the duration, and 1 (corresponding to 1 min) is the discretization step. For the sake of simplicity all the events in this story have the same step. Arcs represent either a compatibility constraint or an activity constraint (case of arcs with diamond) between variables. The compatibility constraint is denoted by one or more qualitative relations (in case it involves at least one composite variable). The activity constraint shows the condition to be satisfied and the qualitative relation between the two variables in case the condition is true. Each qualitative relation is a disjunction of some Allen primitives (Allen, 1983). For example, the relation BM between *John_Pick_Lisa* and *John_Lisa* denotes the disjunction *Before* $\vee$ *Meets*.

In Mouhoub and Sukpan (2005a, 2005b) we have proposed two methods for solving CCTCSPs. These two methods are respectively based on constraint propagation and stochastic local search. The goal of the constraint propagation method is to overcome, in practice, the difficulty due to the exponential search space of the possible TCSPs generated by the CCTCSP to solve and also the search space we consider when solving each TCSP. Indeed, a CCTCSP represents D^M possible TCSPs where D is the domain size of the composite variables and M the number of composite variables. In the same way as reported in Mittal and Falkenhainer (1990) and Sabin, D., Freuder, E. C., and Wallace, R. J. (2003), we use constraint propagation in order to detect earlier later failure. This will allow us to discard at the early stage any subset containing conflicting variables. The method based on constraint propagation is an exact technique that guarantees a complete solution. The method suffers however from its exponential time cost as shown in Mouhoub and Sukpan (2005a, 2005b). In many real-life applications where the execution time is an

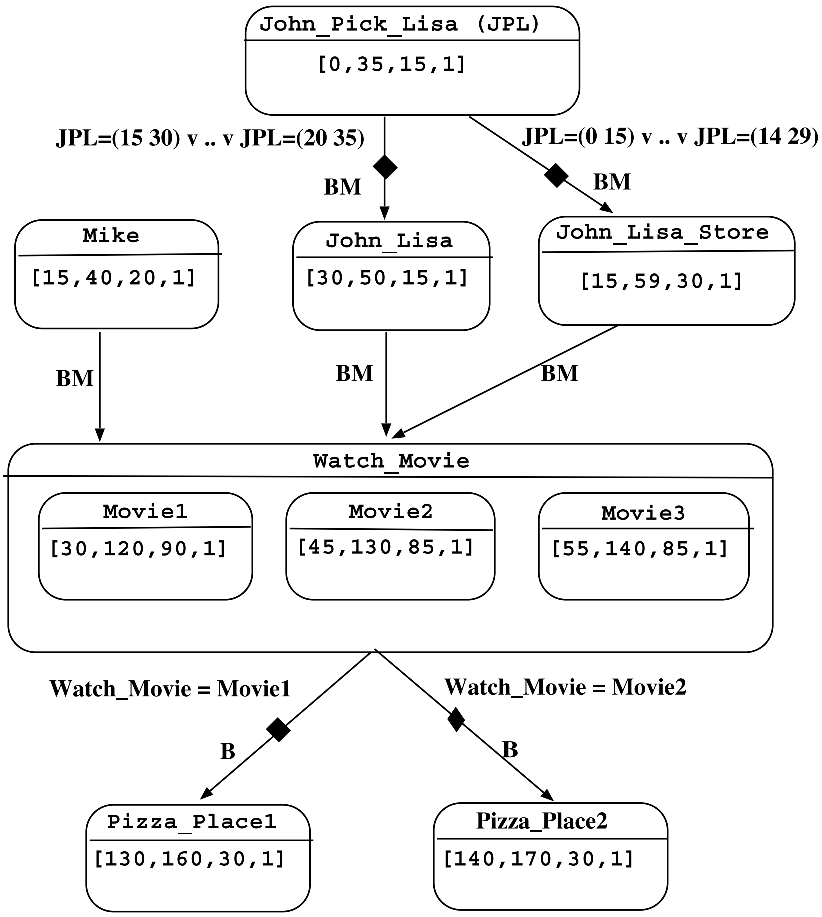

Figure 1. CCTCSP representing Example 1.

issue, an alternative will be to trade the execution time for the quality of the solution returned (number of solved constraints).

This can be done by applying approximation methods such as local search and where the quality of the solution returned is proportional to the running time. In Mouhoub and Sukpan (2005a, 2005b) we studied the applicability of a local search technique based on the Min-Conflict-Random-Walk (MCRW) (Selman & Kautz, 1993a) algorithm for solving CCTCSPs. MCRW has already been applied to solve TCSPs (Mouhoub, 2004). Basically, the method consists of starting from a complete assignment of temporal intervals to events and iterating by improving at each step the quality of the assignment (number of solved constraints) until a complete solution is found or a maximum number of iterations is reached. Experimental study we conducted, in Mouhoub and Sukpan (2005a, 2005b), on randomly generated CCTCSPs demonstrates the efficiency of our exact method based on constraint propagation in the case of middle constrained and over constrained problems while the SLS-based method is the technique of choice for underconstrained problems and also in case we want to trade search time for the quality of the solution returned (number of solved constraints).

3. RELATED WORK

Managing preferences has been extensively studied in the past decade. The CP-net framework (Apt, Rossi, & Venable, 2005; Boutilier, Brafman, Hoos, & Poole, 1999; Boutilier, Brafman, Domshlak, Hoos, & Poole, 2004) is a model for qualitative and conditional preferences under *ceteris paribus*. Preferences are represented separately from hard constraints. *Lexicographically ordered CSP* in Freuder, Wallace, and Heffernan (2003) is an another alternative framework for preferred variables and values. In this latter model, variable selection is the primary factor while value assignment is secondary. Recently, this framework has been extended to *Conditional lexicographic CSP* (Wallace, 2005) for conditional preferences. Finally, quantitative preferences are modeled as a set of soft constraints in Bistarelli, Montanari, and Rossi (1995, 1997a) and Schiex, Fargier, and Verfaillie (1995) supporting different kinds of soft constraints including fuzzy CSPs, weighted CSPs and partial CSPs. These latter frameworks based on semiring structure have been widely used for quantitative preferences in CSPs (Bistarelli et al., 1995, 1997a). A semiring is a tuple $\langle A, +, \times, 0, 1\rangle$ such that:

- A is a set and $0, 1 \in A$;
- $+$, called the additive operation, is a commutative and associative operation such that 0 is its unit element;
- $\times$, called the multiplicative operation, is an associative operation such that 1 is its unit element and 0 is its absorbing element. $\times$ distributes over $+$.

The set of the semiring specifies the values to be associated with each tuple of values of the variable domain. The two semiring operations ($+$ and $\times$) represent constraint projection and combination respectively. A semiring for handling constraints is called *c-semiring*. A *c-semiring* is a semiring with additional properties on the two operations such that $+$ is idempotent, $\times$ is commutative, and 1 is the absorbing element of $+$. A partial order relation $\leq$ is defined over A to compare tuples of values and constraints.

In temporal constraint reasoning, quantitative preferences have been integrated into some existing temporal frameworks (Allen, 1983; Beek, 1992; Dechter, Meiri, & Pearl, 1991; Vilain & Kautz, 1986). Khatib, Morris, Morris, and Rossi (2001) introduced the Simple Temporal Problem with Preferences (STPP). In this latter model, a preference on an interval I is a function with co-domain A (the c-semiring). In Sheini, Peintner, Sakallah, and Pollack (2005) the Disjunctive Temporal Problem (DTP) is extended with preferences using SAT techniques. The Temporal Constraint Network (TCN) (Dechter et al., 1991) is integrated with the addition of a mechanism for specifying preferences, based on the soft constraint formalism (Bistarelli, Montanari, & Rossi, 1997b). In this new model called Temporal Constraint Satisfaction Problem with Preferences (TCSPP), a soft temporal constraint is represented by a pair consisting of a set of disjoint intervals and a preference function: $\langle I = \{[a_1, b_1], \ldots, [a_n, b_n]\}, f\rangle$ where f is defined from I to the c-semiring A. Each feasible solution has a global preference value, obtained by combining the local preference values found. $\times$ is idempotent and also restricts a total order on the elements of A. The c-semiring operations: $+$: $a + b = \max(a, b)$ and $\times$: $a \times b = \min(a, b)$ allow complete solutions to be evaluated in terms of the preference values assigned locally. The optimal solutions to a TCSPP are those solutions that have the best global preference values by the ordering of the values in the c-semiring. Finally, in Badaloni and Giacomin (1999) the 13 basic Allen's relations are assigned with a preference degree, belonging to the interval $[0, 1]$ called IA^{fuz}. IA^{fuz} is closed under *Inverse*, *Conjunctive Combination*, and *Composition*. IA^{fuz} is defined on the set: $I = (r_{1[\alpha_1]}, r_{2[\alpha_2]}, \ldots, r_{13[\alpha_{13}]})$ where $\alpha_i \in [0, 1], r_i \in R$, $i = 1, \ldots, 13$. If α_i is 0, then r_i is an inconsistent relation.

4. NUMERIC, SYMBOLIC, COMPOSITE, AND CONDITIONAL TEMPORAL PREFERENCES

In the following we will define the four types of preferences using the c-semiring structure $\langle A, +, \times, 0, 1\rangle$ for quantitative preferences (Bistarelli et al., 1995, 1997a). Each type of preference is illustrated through the following example (additional information to Example 1).

Example 2. *Lisa prefers John to pick her up early. They prefer to arrive at the theater before the movie starts to get good seats. Lisa prefers to watch movie*$_1$ *to movie*$_2$ *and movie*$_3$*, whereas Mike prefers movie*$_3$ *to movie*$_2$ *and movie*$_1$*. Whoever gets there first will pick the movie that he/she likes.*

4.1. Numeric and Symbolic Preferences

Since the CCTCSPP supports hybrid temporal problems, preference values can be imposed on both numeric and symbolic temporal constraints. Thus, we define two types of soft temporal constraints over the *c-semiring*: *Soft Numeric Temporal Constraint (SNTC)* and *Soft Symbolic Temporal Constraint (SSTC)*.

Definition 2: Soft Numeric Temporal Constraint (SNTC). A Soft Numeric Temporal Constraint (SNTC) is a function $f_{n:e_i} : D_{e_i} \rightarrow A$, where e_i is a temporal event and D_{e_i} its domain of values (time intervals).

In Example 2, the SNTC corresponding to *"Lisa prefers John to pick her up early"* is the function $f_{n:John\_Pick\_Lisa}$ defined as follows.

$$f_{n:John\_Pick\_Lisa}((0\ 15)) = 1.0,$$
$$f_{n:John\_Pick\_Lisa}((1\ 16)) = 0.95,$$
$$\ldots,$$
$$f_{n:John\_Pick\_Lisa}((20\ 35)) = 0.05.$$

Definition 3: Soft Symbolic Temporal Constraint (SSTC). A Soft Symbolic Temporal Constraint (SSTC) is a function $f_{s:c_{ij}} : R_{c_{ij}} \rightarrow A$, where c_{ij} is the symbolic temporal relation between e_i and e_j and $R_{c_{ij}}$ is the set of Allen primitives within c_{ij}.

In Example 2, the symbolic preference *"They prefer to arrive at the theater before the movie starts to get good seats"* favors the Allen relation *before*. Thus, $f_{s:c_{Mike,Watch\_Movie}}(Before)$ has a higher value than $f_{s:C_{Mike,Watch\_Movie}}(Meets)$. Here, we set $f_{s:c_{Mike,Watch\_Movie}}(Before) = 1.0$ and $f_{s:c_{Mike,Watch\_Movie}}(Meets) = 0.6$.

4.2. Composite and Conditional Preferences

A Composite Preference (CompP) is a function $f_{c:x} : D_x \rightarrow A$, where x is a composite variable and D_x its domain of values (events). This function allows us to favor some events within the domain of a given composite variable. The SNTC $f_{n:x_{\downarrow}e}$ of an event e, selected during the backtrack search from the domain of a composite variable x, is recomputed from the composite preference of this latter variable as follows.

Definition 4: Composite Preference (CompP).

Given: a composite variable x,
its domain $D_x = \{e_1, \ldots, e_p\}$,
a composite preference function $f_{c:x}$,
and the selected event e_i,
then: $f_{n:x_{\downarrow}e_i}(I) = f_{c:x}(e_i) * f_{n:e_i}(b_j)$,
where I is a possible time interval of e_i.

A Conditional Preference (CP) allows a preference function (symbolic, numeric or composite) to be added dynamically to the CCTCSPP when a given condition on temporal events or composite variables is true. The condition can be an assignment of particular values to variables.

Definition 5: Conditional Preference (CP). Given a temporal event e (respectively a composite variable x) and a preference function f, a conditional preference has the following form:

$$X_1 \wedge \ldots X_p \overset{condition}{\rightarrow} \text{associate } f \text{ to } e \text{ (respectively to } x)$$

where $X_1, \ldots, X_p$ are temporal variables (composite or events).

Example 3. *The above conditional preference will associate f to e (respectively to x) if condition holds on these variables. condition can be an assignment of particular values to the variables $X_1, \ldots, X_p$. In our Example 2, the conditional preference "Lisa prefers to watch movie$_1$ to movie$_2$ and movie$_3$, whereas, Mike prefers movie$_3$ to movie$_2$ and movie$_1$. Whoever gets there first, will pick the movie that he/she likes." can be formulated by the following two conditional preferences.*

1. *Mike $\wedge$ (John_Lisa $\vee$ John_Lisa_Store) $\overset{condition_1}{\rightarrow}$ assign the composite preference f_1 to the composite variable Watch_Movie.*
2. *Mike $\wedge$ (John_Lisa $\vee$ John_Lisa_Store) $\overset{condition_2}{\rightarrow}$ assign the composite preference f_2 to the composite variable Watch_Movie.*

where:

- *condition$_1$ is: Mike $= I$ and (John_Lisa $= J$ or John_Lisa_Store $= J$) and $end(I) \leq end(J)$*
- *condition$_2$ is: Mike $= I$ and (John_Lisa $= J$ or John_Lisa_Store $= J$) and $end(I) > end(J)$*
- $f_1 = \{movie_3 = 0.9, movie_1 = 0.6, movie_2 = 0.6\}$
- $f_2 = \{movie_1 = 0.9, movie_2 = 0.6, movie_3 = 0.6\}$

4.3. Global Preferences and Optimal Solution to the CCTCSPP

In order to define the global preference of a solution to a CCTCSPP, two other types of preference, namely Associated Local Symbolic Preference (ALSP) and Consistent Binary Assignment Preference (CBAP), are introduced in the following. If C is a symbolic temporal constraint between two events e_i and e_j then the ALSP of C, $f_{as:C}$, can be deduced from the numeric preferences associated to $e_i's$ and $e_j's$ values domain.

Definition 6: Associated Local Symbolic Preference (ALSP).

Given: c_{ij} a constraint between two events e_i and e_j,
$R_{c_{ij}}$ the set of Allen primitives composing c_{ij},
then: for each $r \in R_{c_{ij}}$ such that IrJ

for a given $I \in D(e_i)$ and $J \in D(e_j)$
$f_{as:c_{ij}(r)} = \min(f_{n:e_i}(I), f_{n:e_j}(J))$
where $f_{n:e_i}$ and $f_{n:e_j}$ are the SNTC respectively
for the events e_i and e_j.

A solution to the CCTCSPP is an assignment of numeric intervals to all the temporal events of the problem such that all the compatible constraints are satisfied. The global preference of a solution can be computed by performing the *min* operation on all the Consistent Binary Assignment Preferences. Using the ALSP defined above, a Consistent Binary Assignment Preference (CBAP) is defined as follows.

Definition 7: Consistent Binary Assignment Preference (CBAP).

Given: two events e_i and e_j sharing a constraint c_{ij},
$R_{c_{ij}}$ the set of Allen primitives composing c_{ij},
a CBAP $f_{as:c_{ij}}$,
and a consistent binary assignment
$[e_i = I]\ r\ [e_j = J]$ where:
$r \in R_{c_{ij}}$, $I \in Domain(e_i)$ and $J \in Domain(e_j)$,
$\alpha_i = f_{n:e_i}(I), \alpha_j = f_{n:e_j}(J)$ and $\alpha_r = f_{s:C_{ij}}(r)$

then: $f_{as:c_{ij}(r)} = \min(\alpha_i, \alpha_j)$ and $CBAP(I, J) = \min(f_{as:c_{ij}(r)}, \alpha_r)$

Example 4. *In our Examples 1 and 2, let us assume that during the backtrack search we have made the following decisions (assignments):*

- *Mike* = (15 35), *John_Pick_Lisa* = (0 15), *and John_Lisa_Store* = (15 45)

Using the conditional preferences we have seen earlier in Example 3, the preference function f_1 will be assigned to Watch_Movie. Movie$_3$ (denoted by M_3 in the following) will then be the first value chosen for Watch_Movie. The SNTC of M_3 and the ALSP of Mike and M_3 events with the relation B will be computed as follows:

$$\begin{aligned} f_{n:Watch\_Movie_{\downarrow}M_3}(55\ 140) &= f_{n:Watch\_Movie}(M_3) * f_{n:M_3}(55\ 140) \\ &= 0.9 * 1 = 0.9 \end{aligned}$$

$$f_{as:(Mike,M_3)}(B) = \min(f_{n:Mike((15\ 35))}, f_{n:M_3}(55\ 140)) = \min(1, 0.9) = 0.9$$

The CBAP of the time intervals assigned to Mike and M_3 will then be computed as follows.

$$\begin{aligned} CBAP((15\ 35), (55\ 140)) &= \min(f_{as:(Mike,M_3)}(B), f_{s:(Mike,M_3)}(B)) \\ &= \min(0.9, 1) = 0.9 \end{aligned}$$

Note that since there are no SNTCs defined for the events *Mike* and M_3, the corresponding functions have values 1 for all their elements. The same can be said about the symbolic relation between *Mike* and M_3.

Definition 8: Global Preference (GP). A Global Preference (GP) of a solution $s = \{I_1, I_2, \ldots, I_n\}$ to a CCTCSPP is computed as follows.

Given:
a set of consistent assignments $ca = \{(I_i, I_j)$ such that
$i, j \in n$ and there is a constraint between e_i and $e_j\}$,
Then:
$GP(s) = \min\{CBAP(I, J)$ where $(I, J) \in ca\}$

Definition 9: Optimal Solution (Opt). An Optimal Solution (Opt) of a given CCTCSPP P is the solution having the highest global preference degree.

Given: a CCTCSP P and a set of solutions $S = \{s_1, \ldots, s_n\}$
then: $Opt(P) = \max\{GP(s_1), \ldots, GP(s_n)\}$

5. SOLVING CCTCSPPs

Branch and Bound is a well known method for solving optimization problems. In the case of CCTCSPPs this algorithm is applied to find the optimal solution as follows.

Step 1. The method starts with an initial problem containing a list of initially activated temporal events and composite variables. In order to ensure that domain values are considered according to their preference functions, all the values within each domain are sorted in decreasing order of their SNTC or CompP values (depending whether they belong to an event or a composite variable domain). Similarly, Allen primitives are sorted within their symbolic relations in decreasing order of their SSTC values. Arc consistency is then applied on the initial temporal events and composite variables in order to reduce some inconsistent values which will reduce the size of the search space. If the temporal events are not consistent (in the case of an empty domain) then the method will stop. The CCTCSPP is inconsistent in this case.

Step 2. Following the forward check principle (Haralick & Elliott, 1980), pick an active variable v, assign a value to it and perform arc consistency between this variable and the non assigned active variables. If one domain of the non assigned variables becomes empty then assign another value to v or backtrack to the previously assigned variable if there are no more values to assign to v. Activate any preference function (through conditional preference) and any variable v' (through activity constraint) resulting from this assignment and perform arc consistency between v' and all the active variables. If arc inconsistency is detected then deactivate v' and choose another value for v (since the current assignment of v leads to an inconsistent CCTCSPP). If v is a composite variable then assign an event to it. Basically, this consists in replacing the composite variable with one event *evt* of its domain. We then assign a value to *evt* and proceed as shown before except that we do not backtrack in case all values of *evt* are explored. Instead, we will choose another event from the domain of the composite variable v or backtrack to the previously assigned variable if all values (events) of v have been explored. This process will continue until all the variables are assigned in which case we obtain a solution to the CCTCSPP. Since we are looking for the highest global preference degree, the GP value of this solution will be used as a lower bound (LB) of our branch and bound algorithm. Note that anytime a preference function f is activated (added to the CCTCSPP) through a conditional preference, the domain of values of the variable associated to f is sorted according to this latter.

Step 3. The rest of the search space is then systematically explored as follows. Each time the current variable (event or composite) is assigned a value, an overestimation of the GP value of any possible solution following this decision is computed and used as an upper bound (UB). If $UB < LB$ then the current variable is assigned another value or the algorithm backtracks to the previous variable if all the values have been explored. The overestimated GP is the minimum of the CBAPs of all the assigned variables and the estimated CBAPs involving non assigned variables (including those that can be activated during the remaining search process). An estimated CBAP involving a non assigned variable X_i is calculated as follows.

If the other variable X_j involved by the CBAP is an assigned variable then the estimated CBAP is the minimum of the following:

- the SNTC of the value assigned to X_j,
- the maximum of the SSTCs of all the Allen primitives within the symbolic relation between X_i and X_j,
- and the maximum of the SNTCs of all the values belonging to X_i's domain.

Else (X_j is not assigned yet):

- the maximum of the SNTCs of all the values belonging to X_j's and X_i's domains,
- and the minimum of the SSTCs of all the Allen primitives within the symbolic relation between X_i and X_j.

Note that the arc consistency in steps 1 and 2 above is enforced as shown in the four cases below. We will assume in the following that evt_1 and evt_2 are two events while x_1 and x_2 are two composite variables.

1. **The constraint is** (evt_1, evt_2). Arc consistency (Mackworth, 1977) is applied here i.e., each interval a of evt_1 should have a support in the domain of evt_2.
2. **The constraint is** (x_1, evt_1). Each interval a, from the domain of a given event evt within x_1, should have a support in the domain of evt_1.
3. **The constraint is** (evt_1, x_1). Each interval a, from the domain of evt_1, should have a support in at least one domain of the variables within x_1.
4. **The constraint is** (x_1, x_2). Apply case 2 between x_1 and each interval evt within x_2.

The pseudo code of the arc consistency algorithm based on the above rules is presented in Figure 2. This algorithm is an extension of the well known AC-3 procedure (Mackworth, 1977).

6. EXPERIMENTATION

In order to evaluate the method we propose, we have performed experimental tests on randomly generated consistent CCTCSPPs.

The experiments were performed on a PC Pentium 4 computer running Linux. All the procedures are coded in C/C++. CCTCSPPs are built from TCSPs randomly generated by the model RB proposed in Xu and Li (2000). This model is a revision of the standard Model B (Gent, MacIntyre, Prosser, Smith, & Walsh, 1998; Smith & Dyer, 1996), has exact phase transition and the ability to generate asymptotically hard instances. Following the model RB, we generate each TCSP instance in two steps as shown below and using the parameters n, p, α and r where:

- n is the number of events,
- p $(0 < p < 1)$ is the constraint tightness which can be measured, as shown in Sabin and Freuder (1994), as the fraction of all possible pairs of intervals from the domain of two events that are not allowed by the constraint,
- and r and α $(0 < \alpha < 1)$ are two positive constants.

1. Select with repetition $rn \ln n$ random constraints. Each random constraint is formed by selecting without repetition 2 of n events.
2. For each constraint we uniformly select without repetition pd^2 incompatible pairs of intervals from the domains of the pair of events involved by the constraint. $d = n^\alpha$ is the domain size of each event.

```
REVISE(D_i, D_j)
   REVISE ← false
   for each value a ∈ D_i do
      if not compatible(a,b) for any value b ∈ D_j then
                        remove a from D_i
                        REVISE ← true
      end if
   end for
   return REVISE

REVISE_COMP (D_i, D_j)
   REVISE_COMP ← false
   if i is an event and j is a composite variable (case 3)
      D ← ∅
      D_tmp ← D_i
      for each event k ∈ D_j do
         REVISE(D_i, D_k)
         D ← D ∪ D_i
         D_i ← D_tmp
      end for
      D_i ← D
      if D_i ≠ D_tmp
         REVISE_COMP ← true
      end if
   end if
   if i is a composite variable and j is an event (case 2)
      for each event k ∈ D_i do
         REVISE_COMP ← REVISE(D_k, D_j)
      end for
   end if
   if both i and j are composite variables (case 4)
      for each event k ∈ D_i do
         REVISE_COMP(D_k, D_j)
      end for
   end if
   return REVISE_COMP

AC − 3 − CCTCSPP
   Given a CCTCSPP ⟨E, D_E, X, D_X, IV, C, Act⟩
   i, j and k are variables defined on D_i, D_j and D_k respectively
   Q ← {(i, j)|(i, j) ∈ C}
   while Q ≠ Nil do
      Q ← Q − {(i, j)}
      if i or j is a composite variable (case b, c or d)
         if REVISE_COMP(D_i, D_j) then
            Q ← ∪{(k,i)|(k,i) ∈ C and k ≠ j}
         end if
      else (both i and j are events (case a))
         if REVISE(D_i, D_j) then
            Q ← ∪{(k,i)|(k,i) ∈ C and k ≠ j}
         end if
      end if
   end while
```

Figure 2. AC-3 for CCTCSPPs.

Each CCTCSPP instance is then generated as follows using the parameters N, D, I and a which respectively denote the number of composite variables, their domain size (number of events within each composite variable), the percentage of variables that are initially active and the density of activity constraints.

1. Randomly generate a TCSP with the parameters n, p, α and r as shown above. Symbolic preference values (randomly picked from [0..1]) are then associated to each Allen primitive within each constraint (disjunctive relation). Similarly, random numeric preference values chosen from [0..1] are associated to each event domain value.
2. Generate N composite variables each containing D events and associate to each of these events a random number from [0..1]. This number corresponds to the composite preference associated to the event.
3. Select with repetition $r[(n+N)\ln(n+N)-n\ln n]$ new random constraints (between the $n+N$ composite variables and events), each formed by selecting without repetition 2 of the $n+N$ variables. This will guarantee that the total number of constraints is $r(n+N)\ln(n+N)$ (as per the requirements of the RB model). Each selected constraint C_{ij} involving two variables X_i and X_j is then generated following one of the procedures below. Random symbolic preferences are then associated to each of these constraints in the same manner as shown in step 1 above.
   (a) If both X_i and X_j are events then we uniformly select without repetition pd^2 incompatible pairs of intervals from the domains of X_i and X_j.
   (b) If X_i is composite and X_j is an event (or vise versa) then the constraint will be a disjunction of D relations between the event X_j and each event within X_i domain. Each of these D relations will be generated as shown above in (a).
   (c) If both X_i and X_j are composite then the constraint will be a disjunction of D^2 relations between the pair of events from X_i and X_j domains. Each of these D^2 relations will then be generated as shown above in (a).
4. Select $I(n+N)$ initial variables from $n+N$ $(0 < I < 1)$.
5. Select $a(nd+ND)$ activity constraints for each of the $n+N-I(n+N)$ none initial variables $(0 < a < 1)$. Note that the activity constraints are defined here as $X_i = val \rightarrow X_j$ (where val is a value of X_i's domain) which is less general then the definition we have provided in Section 2. The total number of possible activity constraints is thus equal to $nd+ND$.

As demonstrated in Xu and Li (2000), when the number of variables approaches infinity the phase transition occurs when the constraint tightness $p = 1 - e^{-\frac{\alpha}{r}}$. Thus the phase transition is an asymptotic phenomenon since we can have sharp phase transitions only for infinite number of variables. In addition, the number of variables and constraints of the possible CSPs, each

CCTCSPP contains, is slightly different from the one of the CCTCSPPs from which they are generated.

Although we mentioned in the previous section that we use the forward check principle during search, we consider here other propagation strategies as well. More precisely we compare the following four strategies.

Forward Check (FC). This is the strategy we have described in the previous section (in Step 2).

Maintaining Arc Consistency (MAC). This strategy maintains a full arc consistency on the current and future active variables (variables not yet assigned).

FC+. Same as FC except that the applicability of the arc consistency is extended to non active variables as well.

MAC+. Same as MAC except that the applicability of the arc consistency is extended to non active variables as well.

The CCTCSPP instances are generated with the following parameters: $n = 140$, $N = 10$, $D = 5$, $\alpha = 0.8$, $I = 0.8$, $a = 0.2$ and $r = 0.6$. As mentioned earlier, the phase transition can be computed as follows: $p = 1 - e^{-\frac{\alpha}{r}} = 1 - e^{-\frac{0.8}{0.6}} = 0.73$. In practice the tightness is around 0.7 as we can see in Figure 3. For each test (corresponding to a particular tightness value

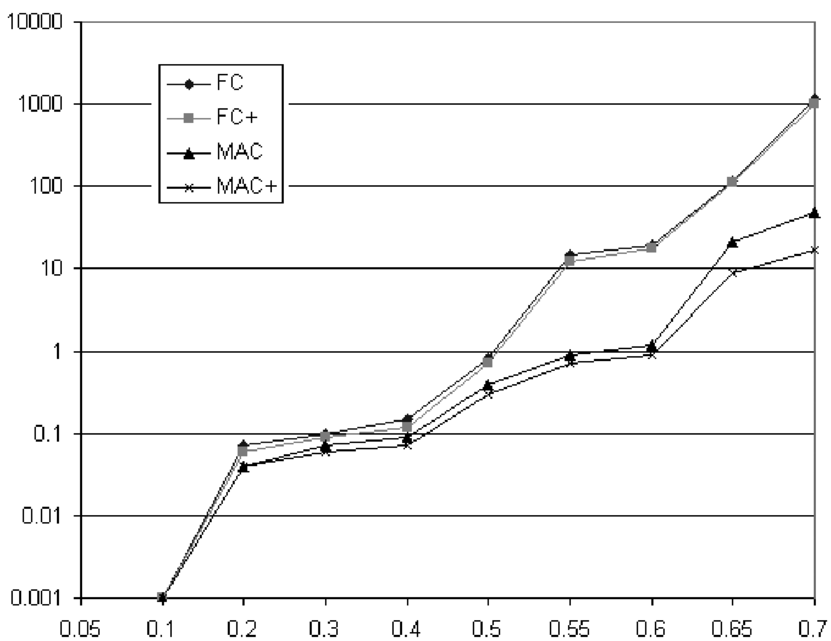

Figure 3. Comparative tests on random CCTCSPPs.

p), each of the four methods is executed on 100 instances and the average running time in seconds is taken.

As we can easily see in Figure 3, for under and middle constrained problems (tightness $\leq$ 0.4) the four strategies have similar running time. However in the case of highly constrained problems the time effort spent by MAC and especially MAC+ starts to pay off. Indeed when we reach the phase transition MAC+ is almost 100 times faster than FC and FC+.

7. CONCLUSIONS

In this paper we have proposed a new framework managing preferences at different levels of the temporal constraint network and in a dynamic environment. This framework is very appealing for a wide variety of real world applications such as reactive scheduling and planning, logistics and temporal databases. The approach we adopted consists in converting a given temporal scenario involving numeric and symbolic time information into a hybrid temporal constraint network where conditional constraints and composite variables are used to add new information (variables and their related constraints) to the constraint network in a dynamic manner during the resolution process. Preferences are associated to numeric, symbolic and conditional constraints as well as composite variables, in order to favor some solutions to the temporal scenario. Finding the best solution is carried out by a variant of the branch and bound algorithm we propose. In order to evaluate the time performance of our solving method, we conducted preliminary tests comparing different propagation strategies on randomly generated CCTCSPPs. The results favor a variant of MAC over the other strategies (Haralick & Elliott, 1980). In the near future, we intend to conduct more experimental study on real-life applications under time constraints with preferences. Another perspective is to consider approximation methods such as Stochastic Local Search (SLS) (Selman & Kautz, 1993b), Genetic Algorithms (GAs) (Craenen & Eiben, 2003) and Ant Colony Algorithms (ACAs) (Stützle & Hoos, 1998). Although these techniques do not always guarantee an optimal solution to the problem, they are very efficient in time (comparing to branch and bound) and can thus be useful if we want to trade the optimality of the solution for the time performance.

REFERENCES

Allen, J. (1983). Maintaining knowledge about temporal intervals. *CACM, 26*(11), 832–843.

Apt, K. R., Rossi, F., & Venable, K. B. (2005). CP-nets and Nash equilibria. In S. deGivry and W. Zhang (Eds.), *Workshop on Preferences and Soft Constraints* (pp. 1–16), Stiges, Spain.

Badaloni, S., & Giacomin, M. (1999). A fuzzy extension of Allen's Interval Algebra. In E. Lamma & P. Mello (Eds.), *Proc. of the 6th Congress of* *the Italian assoc. for artificial intelligence*, Bologna, Italy (pp. 228–237). Springer-Verlag.

Beek, P. van. (1992). Reasoning about qualitative temporal information. *Artificial Intelligence, 58*, 297–326.

Bistarelli, S., Montanari, U., & Rossi, F. (1995). Constraint solving over semirings. In C. Mellish (Ed.), *IJCAI'95: Proceedings International Joint Conference on Artificial Intelligence*. Montreal.

Bistarelli, S., Montanari, U., & Rossi, F. (1997a). Semiring-based constraint satisfaction and optimization. *Journal of the Association for Computing Machinery 44*(2), 201–236.

Bistarelli, S., Montanari, U., & Rossi, F. (1997b). Semiring-based constraint satisfaction and optimization. *J. Association for Computing Machinery 44*(2), 201–236.

Boutilier, C., Brafman, R., Hoos, H., & Poole, D. (1999). Reasoning with conditional ceteris paribus statements. In *Proc. of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Stockholm, Sweden (pp. 71–80) Morgan Kaufmann.

Boutilier, C., Brafman, R. I., Domshlak, C., Hoos, H. H., & Poole, D. (2004). Preference-based constrained optimization with CP-nets. *Computational Intelligence, 20*(2), 137–157.

Craenen, B., & Eiben, A. (2003). Comparing evolutionary algorithms on binary constraint satisfaction problems. *IEEE Transactions on Evolutionary Computation, 7*(5), 424–444.

Dechter, R., Meiri, I., & Pearl, J. (1991). Temporal constraint networks. *Artificial Intelligence, 49*, 61–95.

Freuder, E., Wallace, R. J., & Heffernan, R. (2003). Ordinal Constraint Satisfaction. In *Fifth International Workshop on Soft Constraints (SOFT'02)*.

Gent, I., MacIntyre, E., Prosser, P., Smith, B., & Walsh, T. (1998). *Random constraint satisfaction: Flaws and structure*. (Tech. Rep. No. APES-08-1998). APES Research Group.

Haralick, R., & Elliott, G. (1980). Increasing tree search efficiency for Constraint Satisfaction Problems. *Artificial Intelligence, 14*, 263–313.

Khatib, L., Morris, P., Morris, R. A., & Rossi, F. (2001). Temporal constraint reasoning with preferences. In *International Joint Conference on Artificial Intelligence* (pp. 322–327).

Mackworth, A. K. (1977). Consistency in networks of relations. *Artificial Intelligence, 8*, 99–118.

Mittal, S., & Falkenhainer, B. (1990, August). Dynamic constraint satisfaction problems. In *Proceedings of the 8th National Conference on Artificial Intelligence* (pp. 25–32). Boston, MA: AAAI Press.

Mouhoub, M. (2004). Reasoning with numeric and symbolic time information. *Artificial Intelligence Review, 21*, 25–56.

Mouhoub, M., & Sukpan, A. (2004). Solving conditional and composite temporal constraints. In *The 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)* (pp. 734–741). Boca Raton, FL. IEEE Computer Society.

Mouhoub, M., & Sukpan, A. (2005a). Constraint propagation versus local search for conditional and composite temporal constraints. In M. R. C. van Dongen (Ed.), *Eleventh International Conference on Principles and Practice of Constraint Programming (CP 2005), Workshop on Constraint Propagation and Implementation* (pp. 63–78). Sitges, Barcelona, Spain.

Mouhoub, M., & Sukpan, A. (2005b). A new temporal csp framework handling composite variables and activity constraints. In *The 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)* (pp. 143–149). Hong Kong. IEEE Computer Society.

Sabin, D., & Freuder, E. C. (1994). Contradicting conventional wisdom in constraint satisfaction. In *Proc. 11th ECAI* (pp. 125–129). Amsterdam, Holland: John Wiley and Sons.

Sabin, D., Freuder, E. C., and Wallace, R. J. (2003). Greater efficiency for conditional constraint satisfaction. *Proc., Ninth International Conference on Principles and Practice of, Constraint Programming (CP 2003), 2833*, 649–663.

Schiex, T., Fargier, H., & Verfaillie, G. (1995). Valued constraint satisfaction problems: Hard and easy problems. In C. Mellish (Ed.), *IJCAI'95 Proceedings International Joint Conference on Artificial Intelligence*, Montreal.

Selman, B., & Kautz, H. (1993a). Domain-independent extensions to GSAT: Solving large structured satisfiability problems. In *The 13th International Joint Conference on Artificial Intelligence*, Chambéry, France. Morgan Kaufmann. (pp. 290–295).

Selman, B., & Kautz, H. A. (1993b). An empirical study of greedy local search for satisfiability testing. In *The 11th National Conference on Artificial Intelligence*, Washington, DC. The AAAI Press and The MIT Press. (pp. 46–51).

Sheini, H. M., Peintner, B., Sakallah, K. A., & Pollack, M. E. (2005). On solving soft temporal constraints using SAT techniques. In *Proceedings of the Eleventh International Conference on Principles and Practice of Constraint Programming* (pp. 607–621). Sitges, Barcelona, Spain. Springer.

Smith, B., & Dyer, M. (1996). Locating the phase transition in binary constraint satisfaction problems. *Artificial Intelligence, 81*, 155–181.

Stützle, T., & Hoos, H. (1998). Improvements on the Ant System: Introducing the MAX-MIN Ant System. *Artificial Neural Networks and Genetic Algorithms*, 245–249.

Vilain, M., & Kautz, H. (1986). Constraint propagation algorithms for temporal reasoning. In *AAAI'86* (pp. 377–382). Philadelphia, PA. The MIT Press.

Wallace, R. J. (2005). Conditional lexicographic orders in constraint satisfaction problems. In *Eleventh International Conference on Principles and Practice of Constraint Programming (CP 2005)*. Sitges, Barcelona, Spain (pp. 134–146).

Xu, K., & Li, W. (2000). Exact phase transitions in random constraint satisfaction problems. *Journal of Artificial Intelligence Research, 12*, 93–103.