Generating Scenarios for Periodic Events with Binary Constraints

Lina Khatib
Robert A. Morris
Florida Institute of Technology
Melbourne, FL 32901
email:{lina,morris}@cs.fit.edu

Abstract

Reasoning with repeating events differs from reasoning about single events primarily in the fact that with the former the available information about aspects such as the number *and* period *of the events may be indefinite. Much useful knowledge about repeating events takes the form of relationships between "successive" occurrences of the same event, or "proximate" occurrences of pairs of repeating events. The formulation presented here focuses on such knowledge. A backtracking algorithm for solving reasoning problems involving repeating events is presented and its complexity evaluated.*

1 Introduction

The time associated with repeating events can be represented simply as a set of intervals of time with a temporal structure. Here, we consider a temporal structure which is satisfied by any finite number of non-overlapping intervals. Thus, the plural noun phrase **faculty meetings** stands for a repeating event, and the temporal component of this event can be viewed as a finite set of non-overlapping time intervals. The times assigned to these intervals might be known, either directly or through inference from other knowledge, or at least partially unknown. If unknown, it might be part of a specification of a reasoning problem, whose goal is the assignment of a temporal structure (set of times) to the event.

There are two features which, in our opinion, create contexts for reasoning which do not arise in reasoning about singular events. First, there is the need to explicitly represent information about the *number* of events in a recurring event. Secondly, there is the need to explicitly represent information about the *period* of the recurring event. This period can be expressed in a number of ways. First, it can be expressed by fixing it to a calendar event, for example *twice a week*. Second, it can be expressed in terms of gaps between successive occurrences, e.g. *never on consecutive days*. Third, the period can be expressed in terms of a generalization of a temporal relation, e.g. *only follows*.

The remaining sections of this paper describe a framework for representing information about repeating events in the form of constraints in a CSP. This work builds upon previous work by the same authors, and contains the same representation of single repeating events as that found in the companion paper to this one [Morris and Khatib, 1999]. The focus of this paper is representing binary constraints between repeating events, and using those constraints to transform a specification into a temporal CSP, which can then be solved by the methods described in [Dechter *et al.*, 1991].

2 Binary diagonal temporal relations

Consider the pair of finite, repeating events in Figure 1. By comparing one end-point of I with one of J, information about the relative ordering of these events can be extracted. We refer to this as information about the *relative profile* of two repeating events. For example, the *relative period* profile $p_{s,s}$ displays distances between all pairs of start-points of I and J. For this example, the period profile can be represented as a 4×2 matrix:

$$p_{s,s}[I,J] = \begin{vmatrix} 6 & 26 \\ -1 & 19 \\ -15 & 5 \\ -24 & -4 \end{vmatrix}$$

The value of the ith row and jth column of this array is the difference between the start time of the ith occurrence of I and the jth occurrence of J; positive (negative) values indicate that the J occurrence is after (before) the I occurrence. Different profiles can be examined by taking different combinations of start- and end-points of the repeating events. For example, a relative gap profile $g_{e,s}$ describes the duration of every gap between one sub-interval of I and one of J. In addition, there is a *qualitative* profile which lists all the Allen relations found in the occurrences. These can be depicted in the following

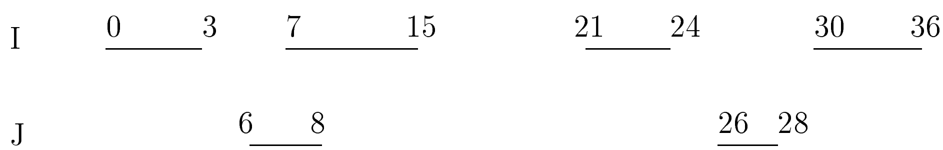

Figure 1: An instantiation of a pair of finite repeating events

matrix format:

$$r_{I,J} = \begin{vmatrix} b & b \\ oi & b \\ bi & b \\ bi & bi \end{vmatrix}$$

Profiles have criteria for *admissibility*, based on the structural assumption that a repeating event is a finite set of disjoint sub-intervals. Thus, rows in quantitative relative profiles must increase in value from left to right, and column values must decrease when read from top down. Admissible qualitative profiles have three segments: a bottom-left triangle of *bi* relations, an upper-right triangle of *b* relations, and a middle region where any of Allen interval relations are allowed.

In specifying patterns of repeating events, a natural constraint to impose is that a certain relationship between occurrences should periodically take place. Thus, the example in the figure satisfies the constraint *the start of each J should be between 2 and 5 times units after the start of someI.* This relationship can be formulated in first-order logic as the period constraint

$$(C) \quad \forall J_i \in J \; \exists I_k \in I \; s(J_i) - s(I_k) \in [2,5],$$

where $s(I_j)(e(I_j))$ is the start (end) of sub-interval I_j. Because constraints of this form tend to be about values in or around the diagonal of a relative profile, they will be referred to as *diagonal constraints.*

A constraint such as (C) can be decomposed into three separate semantic components: first, a pair of quantifiers defining a *mapping* between elements of I and J; second, a pair of end-points $\langle p_1, p_2 \rangle$, where $p_1, p_2 \in \{start, end\}$, designating which sets of end-points from I and J are to participate in the mapping; and third, an interval representing the range of values of temporal distances between points. There are also a qualitative version of a diagonal relation in which the distance interval and point pairs (i.e., the second and third parts of the constraint) are replaced by an Allen relation.

There are many kinds of mappings between pairs of finite sets, based on whether the mapping is partial or total with respect to its domain, whether it is functional or relational (i.e., whether a single domain element can be mapped to more than one element in the range), whether the mapping is into or onto with respect to the range, and whether the mapping is many-to-one or one-to-one. We have used the term *correlation* to describe a mapping between time units of repeating repeating events. Thus, a *functional* mapping restricts the end-point from I to be associated with a single end-point from J. For example, a doctor might wish to imply that a *single* event of administering a drug is to occur between 1 and 3 hours after each therapy session. This implies a functional mapping between therapy sessions and drug administerings. The *each* in the phrase *each therapy session* further implies that this mapping is total. There are also many kinds of partial mappings possible; the least constrained is represented by the quantifier *sometimes*, but there are others, such as the logically infamous *every nth time.* Finally, there are restrictions on the mapping that result from the fact that the relations are diagonal. Formally, this means that for any pair of repeating events, I and J, if there is a diagonal relation $R_{I,J}$ between them, then for any mapping *cor*, if $cor(I_j) = J_k$ and $p > j$, then if $cor(I_p) = J_m$, then $m \geq k$. We call this the *no crossover* restriction. In what follows, all mappings are assumed to be one-to-one and onto; therefore, it is not necessary to introduce syntax for this semantic component of diagonal constraints.

A diagonal constraint such as (C) will be abbreviated as

$$2 \leq p_s[I,J] \leq 4.$$

p_s will be called an *aspect variable*, and can be viewed as ranging over profile values. The companion to this paper [Morris and Khatib, 1999] introduces a language for specifying constraints on single repeating events. Such a specification is a pair $(V, \mathcal{C})$, where V is a set of aspect variables dealing with the number and duration between pairs of end-points of sub-intervals of a single repeating event, and $\mathcal{C}$ is a set of constraints on variables in V. Each constraint is of the form $l \leq v \leq u$, where $v \in V$, and l and u are integers specifying the bounds of the value of v. To this framework, we add a language for specifying constraints between pairs of repeating events, and the notion of a concretization of a binary constraint as the result of introducing a mapping between end-points of sub-intervals in I and J.

A *concretization* of a set of repeating events is a simple temporal network of a Temporal CSP [Dechter *et al.*, 1991]. An example of a set of constraints and a concretization is the following (Example 1), with the graphical visualization found in Figure 2.

Example 1 Specification of constraints on repeating events I and J:

- $2 \le n_I \le 5$; $1 \le n_J \le 2$ (constraints on the number of sub-intervals of I and J)
- $3 \le d_I \le 5$; $3 \le d_J \le 15$ (constraints on the duration of each sub-interval of I and J)
- $3 \le g_I \le 4$ (constraint on the duration of the gaps between consecutive occurrences of I)
- $10 \le e_I \le 50$; $10 \le e_J \le 20$ (constraints on the duration between the start of the first sub-interval of I (J) and the end of the last; this is referred to as the *extent* of a repeating event)
- $2 \le p_{s,s}[I, J] \le 4$ (constraint on the duration between start times of I and J)

In the figure, there are nodes for two repeating events, I and J, with a looping arc labeled by the conjunction of all the constraints specified for these events. A single arc connects the two nodes, labeled by the binary constraint. The two boxes at the bottom are concretizations of the two repeating events, "triggered" by an assignment to the number variables n_I, n_J of each event. That assignment is the label on the arc from the repeating event nodes to the concretization. There is also a concretization of the binary relation between I and J, also labeled by its trigger. A trigger for a binary relation between repeating events is the correlation mapping (cor), which maps relevant end-points of I to those of J. A concretization of a relation adds as many binary relationships between end-points as is required based on the numbers of intervals assigned during the event concretizations. Since J was concretized into one interval, there is only one arc in the concretization of the relation. The label shows the mapping selected. Another concretization of the same relation would have assigned $cor(I_1) = J_1$. Finally, notice that in the concretization of J, consisting of a single sub-interval, the extent of the recurring event is the same as the sub-interval duration. Thus, the two constraints for sub-interval and extent were replaced by a single constraint consisting of the intersection of the two intervals. If this intersection were empty, the concretization would be inconsistent.

3 Solving for scenarios of repeating events

Given a set of repeating events, with constraints defined on each, we consider the problem of generating a single *scenario* which satisfies all the constraints. We call such a problem a Repeating-Event CSP (RE-CSP). A scenario is a consistent concretization of all the repeating events and their binary relations.

We speculate, but are not in the position to prove here, that the representation of repeating events and their binary relations presented above is powerful enough that any complete algorithm that guarantees consistency on the temporal information is NP-complete (there is effort currently to finding a proof of this claim). This despite the fact that the representation is restricted in a way that ensures that all concretizations define STPs, consistency of which can be detected in polynomial time. The reason for this claim is the cost of generating consistent concretizations of the binary relations, which requires examining the set of all possible mappings between finite sets in order to find correlations in the worst case. In this section, we examine in detail the cost of applying a worst-case exponential algorithm, based on backtracking, for generating a single set of assignments to all the variables used in specifying a set of repeating events. We call such a problem a Repeating Event Temporal CSP (RE-TCSP).

The following mapping transforms a formulation of a RE-CSP into one that is solvable by chronological backtracking (BT), the standard method for solving CSPs. A BT algorithm incrementally instantiates CSP variables, testing the result against the relevant constraints, until either all the variables have been instantiated or no solution was determined to exist. In a RE-CSP solver, variable instantiation is re-interpreted as finding a consistent concretizing of a single repeating event, using the algorithm described in [Morris and Khatib, 1999]. Second, the consistency checking phase of BT corresponds in a RE-CSP to finding consistent concretizations of all binary relations associating the most recently concretized event with each of the previously concretized events. From a consistent STP network in which all repeating events and their relations have been concretized (if such a network exists), a scenario can be generated.

Below are pseudo-code for algorithms which mimic the basic operations of BT. The input to the main function, BT_RE, is a network RN of repeating events and binary diagonal relations. Its output is a consistent simple temporal network, STN, if one exists, from which scenario tuples can be derived. BT_RE initializes STN and calls BT_RE1, which recursively tries to add concretizations of each repeating event, based on its constraints. This procedure calls $Conc\_R$ (concretize relations) for each binary relation between the currently concretized event I_i and all previously concretized events I_j. $Conc\_R$ assumes an enumeration of all the admissible mappings consistent with the diagonal constraint $R_{i,j}$. Such an enumeration can easily be constructed, e.g., based on the relative ordering of finite sequences σ, such that $\sigma(i) = p$ if and only if $Cor(I_i) = J_p$. Since concretizations of repeating events are networks, and concretizations of relations are labeled arcs, then it's possible without abusing terminology to add concretizations to existing networks. Thus, in $Conc\_R$, testing consistency of $STN' = (V, E \cup E_m)$, where E_m is a mapping, means testing the consistency of the network that results from adding the set of labeled arcs to STN implied by the mapping. Similarly, where $Conc[I_i, k]$ is the function

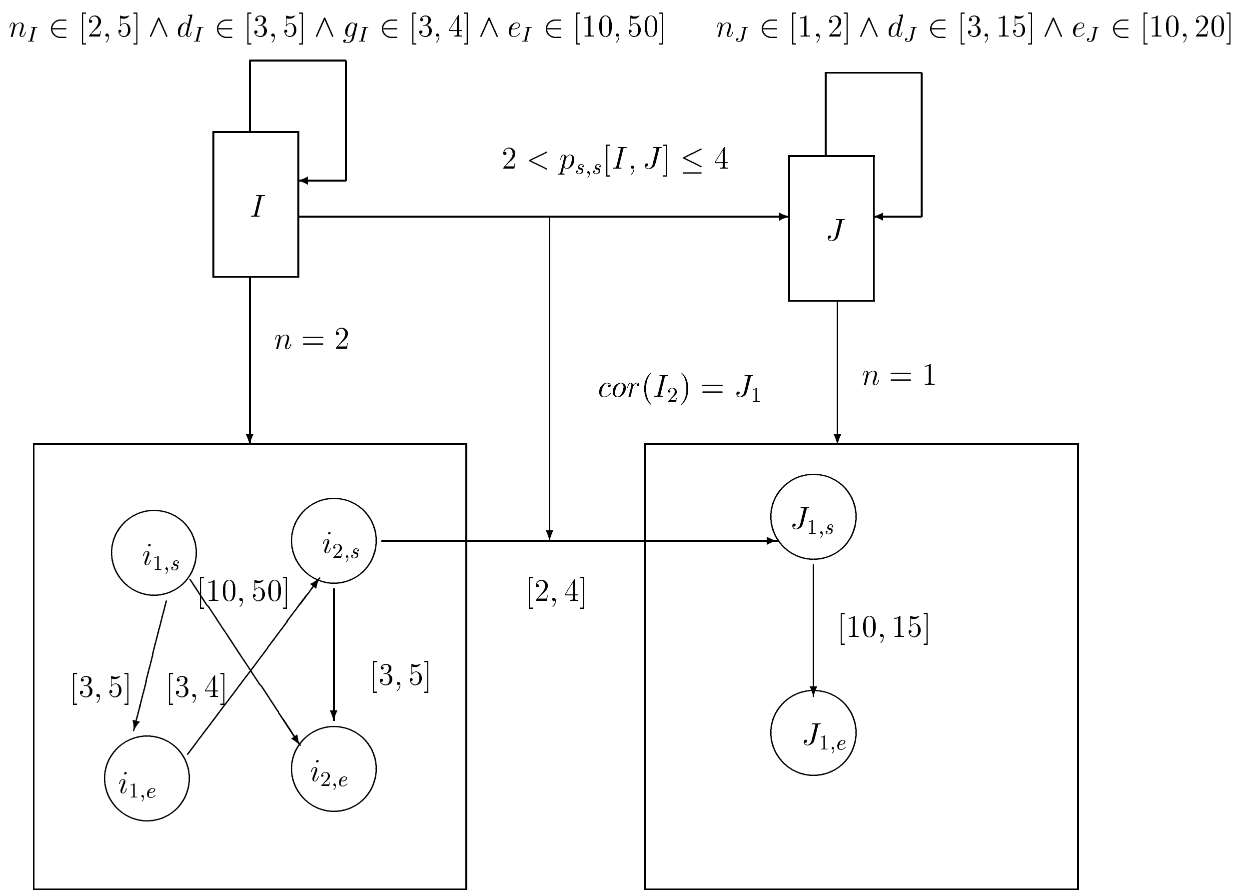

Figure 2: Concretizations of two repeating events and a diagonal relation between them

that builds the network of k sub-intervals of I_i, with appropriate edges between them based on the structural constraints associated with I_i, the meaning of the operation $STN \cup Conc[I_i, k]$ is clear.

function $Conc\_R(R_{i,j}, STN)$
/* $STN = (V, E); R_{i,j} \in R$ */
if $R_{i,j}$ is universal **then** return(true);
for $E_m \in \{$ possible mappings between I_i and $I_j\}$
  if consistent($STN' = (V, E \cup E_m)$)
    $STN = STN'$;
    return(true);
  endif
return(false);
end

function $BT\_RE(RN)$
/* **input**: $RN = (I, R)$ */
/* **output**: a consistent STN, if one exists.*/
$STN = \emptyset$;
$BT\_RE1(RN, STN, I_1, n)$;
return(STN)

procedure $BT\_RE1(RN, STN, I_i, n)$
/* Build a concretization of I_i and its relations */
/* $STN = (V, E)$ is a simple temporal network */
if $i > n$ **then** return(STN);
for $k \in [l_i \ldots u_i]$
/* for each value of number aspect variable */
  if consistent($Conc[I_i, k]$) /* test concretization */
    $STN = STN \cup Conc[I_i, k]$
    /* add concretization of I_i */
    consistent = true;
    j = 1;
    while (consistent and $j < i$)
      /* consider all relations $R_{i,j}, R_{j,i}$ */
      consistent = $Conc\_R(R_{i,j}, STN)$;
      if consistent then
        consistent = $Conc\_R(R_{j,i}, STN)$;
      $j = j + 1$;
    end /* end while */
    if consistent **then**
      $BT\_RE1(STN, I_{i+1}, n)$;
  endif
return(nil); /* no solution found */
end

To measure the worst case complexity of solving RE-CSPs using this method, a number of parameters are involved. These are summarized in the fol-

n	number of repeating intervals
n_{sc}	max_i number of concretizations
n_{rc}	$max_{i,j}$ number of relational concretizations
m	max_i number of sub-intervals
T_{sc}	max_i time for building concretization of I_i
T_{rc}	$max_{i,j}$ time for building relational concretizations
$T_{STP}[i]$	maximum time for testing an STN with i nodes for consistency

Figure 3: Explanation of variables used in complexity analysis

lowing table of variables in Figure 3 . In the table, max_i is to be read "the maximum over all the set of repeating events in the RE-CSP", and $max_{i,j}$ is to be read "the maximum over all pairs of repeating events in the RE-CSP". We now prove the following result:

Theorem 1 Solving for RE-CSPs takes $O(m^{n+m+3}n^5)$ time.

proof. The worst case time complexity can be expressed using the variables in the table above as

$$T_{RE-CSP} =$$

$$n_{sc}^n[T_{sc} + (2m)^3 + \Sigma_{i=1}^n \Sigma_{j=1}^{i-1}(n_{rc}[T_{rc} + T_{STP}[2im]])].$$

The formula states that each possible structural concretization is considered, and tested for consistency. For each concretization of a repeating event I_i, every mapping between end-points of I_i and every other I_j that preceded it in the concretization process is considered, and the result tested for consistency.

Solving the above formula, taking into consideration the following:

- $T_{SC} \leq 4m$ since the process of concretization involves constructing $2m$ nodes, m edges for sub-interval duration, $m-1$ edges for gaps, and 1 edge for extent;
- $n_{sc} \leq m$, the upper bound on the number of sub-intervals;
- $n_{rc} \leq m^m$, the upper bound on the number of mappings between two finite sets of size m; and
- $T_{rc} = m$.

We conclude, substituting into the above formula,

$$T_{RE-CSP} =$$

$$m^n[4m + (2m)^3 + \Sigma_{i=1}^n \Sigma_{j=1}^{i-1}(m^m[m + (2im)^3])].$$

Ignoring constants and insignificant negative terms, this formula can be eventually expressed as

$$O(m^{n+1} + m^{n+3} + m^{n+m+1}n^2 + m^{n+m+3}n^5 + m^{n+m+3}n^4).$$

Observing the dominant term of this formula suffices to prove the theorem.

Although this result indicates that solving RE-CSPs is in the worst case a prohibitively costly undertaking, there are ways of restricting the formulation of the problem that would result in significant savings. First, in real problems there are typically much fewer than m^m admissible mappings implied by a relational constraint. Restricting, for example, mappings to be one-to-one cuts down this number. Secondly, if there are only a few repeating events to be assigned, or a relatively few number of occurrences for each, then the number of concretizations or relational mappings to consider will be small. Third, the RN network may be sparse in the number of relations between repeating events. Then the number of universal binary relations is high, and there is no need to consider large numbers of mappings between sub-intervals. Note that a universal relation between two items means there is no restriction on the relation between them.

4 Discussion

Computational aspects of recurrence and periodicity have been studied in the temporal database literature, in work related to specifying and verifying the correctness of continuously operating concurrent programs such as operating systems, as well as AI. Modal temporal logic forms the framework for much of this work, although classical systems of temporal logic do not allow for many forms of periodic knowledge to be expressed. To attain the needed expressive power, fixed point operators are introduced; some fixed point extensions (e.g., Vardi's USF logic) to propositional modal logic are decidable. Alternatively, second-order extensions to predicate logic have been proposed to formalize the logic of repeating sequences; the system $S1S$ (monadic second-order theory of successor) is decidable, and a decision procedure is available in the form of Büchi automata [Thomas, 1990]. In the AI literature, the interest in repeating events has stemmed primarily from the desire to represent calendar time, [Poesio and Brachman, 1991], or for querying planning or scheduling information. [Koomen, 1991]. A representation of repeating events has proved useful in medical applications such as monitoring the effectiveness of therapy [Keravnou and Shahar, forthcoming]. The approach taken here of clustering intervals and applying structural assumptions about time within the cluster in order to reduce the amount of processing required by the reasoner, is reminiscent of the notion of a reference interval, first introduced by Allen in [Allen, 1983]. The work of Ligozat [Ligozat, 1991] and Ladkin [Ladkin, 1986] on generalized intervals inspired many features of the approach to repeating event representation taken in this paper.

5 Summary

This paper has presented a formulation of repeating events within the CSP framework. In reasoning about repeating events there is the need to manipulate indefinite information about the number of times an event can occur, and the distribution of those occurrences over time. This leads to a language and framework for reasoning in which constraints are formulated about number and distribution. Such abstraction allows for potentially useful problems to be formulated and solved using valued CSPs. Although existing techniques in CSP solving can be applied to reasoning about repeating events, additional sources of complexity arise which threaten the ability to effectively solve these problems. By limiting the expressive power of the constraint language, e.g. by considering only what was referred to here as *diagonal binary relations*, some savings in computational resources are acquired, although the worst-case behavior is still potentially prohibitive for large problems. Other savings can possibly be gained by problem reduction, local search, and parallelism, although no results have been yet obtained which confirm this. The need to manage information about periodic events should, we feel, motivate additional research leading to finding answers to these and other open questions.

References

[Allen, 1983] Allen, J. Maintaining Knowledge About Temporal Intervals. In *Readings in Knowledge Representation*, Brachman, R., and Levesque, H., pages 510-521, Morgan Kaufmann Publishers.

[Dechter *et al.*, 1991] Dechter, R., Meiri, I., and Pearl, J. Temporal Constraint Networks. *Artificial Intelligence*, 49:61-95, 1991.

[Keravnou and Shahar, forthcoming] Keravnou, E. and Shahar, Y. Temporal Reasoning in Medicine. Forthcoming in *Handbook of Time in AI.*

[Koomen, 1991] Koomen, J. Reasoning about Recurrence. *International Journal of Intelligent Systems* 6:461-496, 1991.

[Ladkin, 1986] Ladkin, P. B. Primitives and Units for Time Specification. *Proceedings of AAAI-86*, (San Mateo:Morgan Kaufman), 354-359, 1986.

[Ligozat, 1991] Ligozat, G. On Generalized Interval Calculi. *Proceedings of the Ninth National Conference on Artificial Intelligence* (AAAI-91), 234-240, 1991.

[Morris *et al.*, 1996] Morris, R., Shoaff, W., and Khatib, L. Domain Independent Reasoning about Recurring Events. *Computational Intelligence*, 12(3):450-477, 1996.

[Morris and Khatib, 1999] Morris, R., and Khatib, L. Optimization in Constraint Reasoning about Repeating Events. *Proceedings of TIME-99.*

[Poesio and Brachman, 1991] Poesio, M. and Brachman, R. Metric constraints for maintaining appointments: dates and repeated activities. In *Proceedings of AAAI-91*, pp. 253-259, 1991.

[Thomas, 1990] Thomas, W. Automata on Infinite Objects. *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics.* MIT Press/Elseview, 1990.