

Automatic Generation of Efficient Orderings of Events for Scheduling Applications

Robert A. Morris

Florida Institute of Technology

Computer Science Program

150 W. University Blvd.

Melbourne, FL 32901

phone: (407) 768-8000, ext. 7290 fax: (407) 984-8461

email: morris@cs.fit.edu

KEY WORDS AND PHRASES

Scheduling, similarity-based clustering

PROBLEM STATEMENT

In scheduling a set of tasks, it is often not known with certainty how long a given event will take. We call this *duration uncertainty*. For example, as part of the task of making a telescope observation, the telescope must be accurately centered on a star. The time required to perform this subtask cannot be accurately predicted, since it depends on factors which vary from execution to execution (e.g., the position of the telescope at the start of the execution of this task).

Duration uncertainty is a primary obstacle to the successful completion of a schedule. If a duration of one task is longer than expected, the remaining tasks are delayed. The delay may result in the abandonment of the schedule itself, a phenomenon known as schedule *breakage*. One response to schedule breakage is on-line, *dynamic rescheduling*. A more recent alternative is called *proactive rescheduling* [2]. This method uses statistical data about the durations of events in order to anticipate the locations in the schedule where breakage is likely *prior* to the execution of the schedule. It generates alternative schedules at such sensitive points, which can be then applied by the

scheduler at execution time, without the delay incurred by dynamic rescheduling.

This paper proposes a technique for making proactive error management more effective. The technique is based on applying a *similarity-based method of clustering* to the problem of identifying similar events in a set of events. The remainder of this paper consists of a discussion of the following:

1. The intuitions underlying the technique;
2. The way in which clustering techniques from the AI literature can be applied to the problem of managing duration uncertainty in scheduling;
3. The requisite assumptions about the domain for applying the technique; and
4. An implementation strategy.

INTUITIONS

The set of events under consideration have occurrences which need to be scheduled. The goal is to find an ordering of these occurrences which minimizes the amount of expected duration uncertainty associated with each. The knowledge used to find the ordering comes from observations of repeated past occurrences of the same events. Figure 1 represents a repeated occurrence of an event *E*. *E* recurs 4 times over a stretch of time. Duration uncertainty is depicted visually as the difference in the

E — — — —

Figure 1: A repeating event

lengths of each line representing a single occurrence. We assume that the events under consideration all have the tendency to exhibit duration uncertainty.

The heuristic being formalized here is that duration uncertainty can often be reduced by assigning an event in such a way that it is in temporal proximity to a *similar event*. A mundane example will illustrate. Suppose I am scheduling my daily household chores. I find that I must complete three tasks: clean the kitchen (K), clean the bathroom (B) and work in the garden (G). I can do these in any order; my main constraint is to finish all three within a certain time frame. One is clearly led to a plan to perform K and B together, either before or after G . Why? The tasks are similar, either in that they are both cleaning tasks, or perhaps also because they are indoor tasks.

How does the act of scheduling similar events in close temporal proximity lead to a reduction of duration uncertainty? Intuitively, actions are sometimes similar because they share a number of *stages*. For example, any cleaning room action consists of a *preparation stage* consisting of getting the mop or broom, getting floor cleaner, water, bucket, etc. If I perform the cleaning room actions together, say $K \rightarrow B$ (clean the kitchen followed by clean the bathroom), the preparation stage of B will not be required (or be simplified). Since the duration of any action is the sum of the durations of its stages, the duration uncertainty of the whole will be a similar function of the duration uncertainty of the different stages. It follows that I should be able to more accurately predict how long the bathroom cleaning will take *when preceded by the kitchen cleaning action* than I could

B — — — —
 K — — — —

Figure 2: Pairing an event with a similar event

predict its duration in isolation, or when preceded by a dissimilar event. This conclusion is justified by noting that the preparation stage, in such a situation, does not exist; hence, trivially, there is no uncertainty associated with it, which reduces the uncertainty of the whole event. Graphically, this can be represented as in Figure 2. This figure represents the expected durations of kitchen events when paired with the similar, bathroom cleaning event. On the other hand, if paired with a dissimilar event (e.g. gardening), one would expect K to behave as in Figure 1.

In ordering mundane events, we implicitly bring to bear the ability to apply concepts which cluster events into similarity classes. This paper addresses the same problem when such *a priori* conceptual knowledge about a domain is lacking. For example, in the telescope scheduling domain, it may be difficult or impossible to classify *a priori* whether two tasks to be scheduled are similar or not. The main contribution of this paper is to suggest that there is *a posteriori* knowledge (knowledge gained from experience) that can be used to infer the similarity of events.

COMPUTATIONAL MODEL

The computational problem to be solved can be stated as follows: given a set E of k events, find an ordering $E_1 \rightarrow E_2 \rightarrow \dots \rightarrow E_k$ of all the elements in E which minimizes the expected duration uncertainty over all members of E . The previous section justified the intuition that some orderings of events will exhibit less duration uncertainty than others. In this

section, a technique for finding these preferred orderings will be presented.

Similarity Based On Relative Durations of Events

Based on observations in the previous section, the notion of similarity between two events e and e' can be induced from observations of the durations of each event when they are placed in close temporal proximity.

Definition 1 The relative duration of e with respect to e' ($rd(e, e')$) is the duration of e when e immediately follows e' . The relative average duration of an event e with respect to an event e' is the average duration of e when immediately followed by e' , over a set of occurrences of e and e' .

$rd(e, e')$ can be viewed as a discrete random variable, associating a duration with the outcome of pairing the two events. Let $\sigma_{rd(e, e')}$ denote the standard deviation of $rd(e, e')$. It is then possible to define the notion of relative similarity between triples of events e_1, e_2, e_3 :

Definition 2 e_1 is at least as similar to e_2 as to e_3 if $\sigma_{rd(e_1, e_2)} \leq \sigma_{rd(e_1, e_3)}$.

An absolute concept of similarity can be defined when a similarity *threshold* is postulated. Let θ be such a threshold. Then:

Definition 3 Let e and e' be events. Then e is similar to e' if $\sigma_{rd(e, e')} \leq \theta$.

Any similarity relation is reflexive, symmetric, and intransitive. The claim here is that comparing the value of $\sigma_{rd(e_1, e_2)}$ to a threshold can be viewed as applying a similarity relation. Clearly, reflexivity and intransitivity are satisfied. By definition, symmetry implies that if $\sigma_{rd(e, e')} \leq \theta$, then $\sigma_{rd(e', e)} \leq \theta$. Reflections from intuition should make this assumption plausible. Recall that the postulated reason for reduction of duration uncertainty when events are paired to similar events is that they share a stage, which is eliminated or

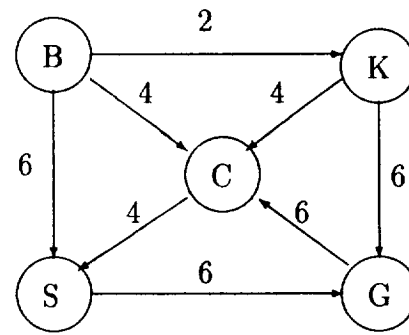


Figure 3: A σ -Graph For Five Events

simplified when the events are paired together. Clearly, the ordering of the pairing is irrelevant. For example, whether $K \rightarrow B$ or $B \rightarrow K$, the duration uncertainty of the later event will be reduced. Hence, it is reasonable to assume that similarity, defined in the previous definition, is symmetric.

Relation to Clustering Methods

In order to reduce duration uncertainty in an error management system for scheduling, events should be ordered in a way that similar events are clustered. The *similarity-based* clustering method [3] is a weak AI method which can be employed to generate efficient orderings. The computational problem of interest here can be viewed as an instance of *one-dimensional clustering*. For such a problem, the goal is to reduce the number of distinct values of a set of variables by identifying near-equivalence classes of values based on similarity. To briefly illustrate the technique of clustering, we introduce a data structure called a σ -graph:

Definition 4 A σ -graph is a weighted directed graph with the following characteristics. Each vertex is labeled by one of the elements in a set E . Each directed edge (e_i, e_j) between source e_i and target node e_j is labeled with a value representing the degree of similarity between e_i and e_j .

To illustrate, consider a slightly more complex mundane example. Now there are

five events, including K , B and G , as before, but also including the tasks *wash car* (C) and *go to store* (S). An incomplete σ -graph for this set of events is found in Figure 3. Here, the lower the value on an arc, the greater the degree of similarity between the two events.

Clustering techniques are traditionally used for automating concept formation. One clustering method (called *agglomeration*), fuses entities to form groupings based on the threshold of minimum similarity. The fusion process stops when all values exceed the threshold. For example, if the threshold is assumed to be 2, the result of the agglomerative process applied to the example would fuse B and K into a cluster. For our purposes, however, clustering is a means to an end, viz., to generate an ordering of events which reduces the amount of duration uncertainty with which a proactive scheduling error manager needs to contend. The following section describes how similarity-based clustering can be implemented for this purpose.

Implementation and Intended Use

The procedure for generating efficient orderings of events based on relative durations is intended to be used as a preprocessing stage in a proactive error management system for scheduling. The stage can be viewed as one that deletes from the set of possible orderings those which exhibit the most duration uncertainty.

Assume as input a set E of k events. The set E has been executed up to m times in some or all of the $k!$ permutations of the orderings of the events in E . Assume an ordering of these permutations and executions. Let $rd(E_i, E_j)[p, q]$ represent the duration of E_i when immediately followed by E_j on the p^{th} occurrence of the q^{th} permutation of E ; thus $1 \leq p \leq m$ and $1 \leq q \leq k!$. This yields a set of $O(k![m(k-1)])$ values of $rd(E_i, E_j)[p, q]$ for

each pair $E_i, E_j \in E$. From this data, an ordering of a set E of events which minimizes duration uncertainty is based on the following steps:

1. For each E_i in E , compute the mean of the set $\{rd(E_i, E_j)[p, q] : 1 \leq p \leq m, 1 \leq q \leq k!\}$, and $\sigma_{rd(E_i, E_j)}$, for each pairing of E_i with other $E_j \in E$;
2. Form a σ -graph with E the set of vertices and for each pair E_i, E_j in E , there is an arc labeled with the value of $\sigma_{rd(E_i, E_j)}$; and
3. Apply an all-pairs shortest-path algorithm [1], such as Floyd-Warshall, to generate an ordering of the events.

For example, assume that Figure 3 represents the result of completing step 2 in the procedure. Thus, the labels on the arcs represent the standard deviations of the relative durations of the event occurrences connected by the arc. If the claims made in this paper are plausible, then such values would be the kind expected, since they reflect the intuitive degree of similarity among the events. Then, the result of applying step three would yield

$$B \rightarrow K \rightarrow C \rightarrow S \rightarrow G$$

as well as other orderings which are minimal with respect to duration uncertainty.

An example of a proactive scheduling system which might benefit from the account presented here is the *Just-In-Case* (*JIC*) error management technique described in [2]. This technique analyzes a schedule of telescope observations for possible execution breaks. For the break point with the highest probability of occurrence, the system forms a contingent alternative schedule. *JIC* utilizes duration uncertainty measures to calculate the possible schedule break points. As a preprocessing stage to the error management procedure, the three stage method presented in this section could be applied to discriminate among different orderings of the events, selecting the ones which minimize

duration uncertainty. This would reduce the amount of anticipated break points with which the error manager has to contend.

ASSUMPTIONS AND LIMITATIONS

To be of optimal benefit for its intended use, the events to be analyzed by the method should possess the following properties:

1. The events in E should be causally independent; this means at least that:
 - No occurrence E_i in E prohibits the execution of any other E_j ; and
 - No occurrence E_i presupposes the execution of some other E_j ;

and

2. Each of the events in E has the tendency to exhibit duration uncertainty; this means that, considered in isolation, the standard deviation of the duration of each event is high.

Even with these minimum assumptions, $\sigma_{rd(E_i, E_j)}$ is a coarse measure of event similarity. For example, assume E_i consists of the stages A , B and C , and E_j consists of A , E , and F . Assume that the duration uncertainty of E_j is caused completely by stage F . Then, the approach proposed here would fail to recognize that the two events are similar (in the sense of sharing a common stage A), since E_j would not demonstrate a reduction of duration uncertainty when paired with E_i . In such a case, it would be useful to view the *absolute* reduction in mean duration as evidence for its similarity to E_i . That is, since E_j shares a stage with E_i , its pairing with E_i should result in a reduction of the time it takes to execute. Hence, it may be the case that both mean duration and standard deviation should be viewed as the measure of similarity. This could be easily added to the implementation by including mean duration as part of the labels on the arcs of the σ -graph. The addition would imply a two dimensional description space for the events,

and a similarity concept based on a *vector* of attributes.

There may be other forms of causal interaction which would make the ordering produced by this procedure less preferred than others.¹ Consider for example events E_i and E_j again. Perhaps the pairing $E_i \rightarrow E_j$ would result in a reduction of the standard deviation of the duration of E_j , and hence be preferred by the proposed model. However, it is possible that this pairing would increase the *absolute* duration of E_j .

CONCLUSION

This paper has offered an approach for aiding proactive error management techniques for scheduling. The idea is to use statistical temporal information about event occurrences to induce similarities among these occurrences, when conceptual information about the same events is unavailable. Pairing similar events in close temporal proximity can often reduce the uncertainty in the expected duration of the events. This leads to the potential for a reduction in the amount of rescheduling required by the proactive error manager.

References

- [1] Cormen, T.; Leiserson, C.; and Rivest, R., 1991. *Algorithms*. Cambridge, Mass.: MIT Press.
- [2] Drummond, M.; Bresina, J.; Swanson, K., 1994. Just-In-Case Scheduling. In *Proceedings of AAAI-94*, forthcoming.
- [3] Michalski, R. S.; and Stepp, R. E., 1987. Clustering. In *Encyclopedia of Artificial Intelligence*, S. Shapiro, ed. Cambridge, Mass: MIT Press.

¹This point was made by one of the reviewers.

